



**Centro Universitário de Brasília
Instituto CEUB de Pesquisa e Desenvolvimento - ICPD**

GLEYSON LUIZ S. CRUZ

**SISTEMA DE DETECÇÃO DE INTRUSÃO UTILIZANDO OPENBSD E
SNORT: MITIGANDO ATAQUES DE NEGAÇÃO DE SERVIÇO**

Brasília
2013

GLEYSON LUIZ S. CRUZ

**SISTEMA DE DETECÇÃO DE INTRUSÃO UTILIZANDO OPENBSD E
SNORT: MITIGANDO ATAQUES DE NEGAÇÃO DE SERVIÇO**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em Redes de Computadores com Ênfase em Segurança.

Orientador: Prof. Dr. José Eduardo M. S. Brandão

Brasília
2013

GLEYSON LUIZ S. CRUZ

**SISTEMA DE DETECÇÃO DE INTRUSÃO UTILIZANDO OPENBSD E
SNORT: MITIGANDO ATAQUES DE NEGAÇÃO DE SERVIÇO**

Trabalho apresentado ao Centro
Universitário de Brasília (UniCEUB/ICPD)
como pré-requisito para a obtenção de
Certificado de Conclusão de Curso de
Pós-graduação *Lato Sensu* em Redes de
Computadores com Ênfase em
Segurança.

Orientador: Prof. Dr. José Eduardo M. S.
Brandão

Brasília, 13 de Novembro de 2013.

Banca Examinadora

Prof. Me. Marco Antônio de Oliveira Araújo

Prof.^a Dra. Tânia Cristina da Silva Cruz

*Aos meus Pais, Sebastião da Cruz e Maria
Isileide S. Cruz. Vocês são a minha inspiração e
exemplos à serem seguidos... Obrigado!*

AGRADECIMENTOS

A Deus, por me abençoar a alcançar meus objetivos, dando-me saúde e forças para prosseguir diariamente, mesmo em meio as dificuldades que surgem. Deus, obrigado por tudo.!

Aos meus pais, Sebastião da Cruz e Maria Isileide S. Cruz, por todo o carinho e paciência. Em todos os momentos, sejam eles bons ou ruins, vocês sempre estão me apoiando e aconselhando. Espero um dia conseguir retribuir à vocês, por todo o sacrifício que tiveram que passar, para que eu pudesse ter e ser alguma coisa hoje. Também agradeço a minha querida irmã, Gabrielly S. Cruz, pelo cuidado e carinho que tem por mim.

Ao meu orientador, Prof. Dr. José Eduardo M. S. Brandão, pelo incentivo, apoio e amizade. Muito obrigado pelos conselhos e pela paciência, foi uma honra tê-lo como orientador.

Ao Prof. Dr. Gilson Ciarallo, do curso de Metodologia Científica. Muito obrigado pela apoio e paciência professor, suas dicas foram de grande ajuda.

Aos meus grandes amigos e colegas de curso, que jamais serão esquecidos, Cleyton Silva, Fabrício Cardoso e Márcio Júnio, que juntos, passamos por dificuldades e conquistas. Gostaria de deixar registrado que foi uma grande honra e uma imensa alegria, conhecer a cada um de vocês. Muito obrigado pela amizade.!

*“Porque assim me disse o Senhor: Vai, põe
uma sentinela, e ela que diga o que vir.”*
Isaías 21:6

RESUMO

Este trabalho apresenta um estudo sobre Sistemas de Detecção de Intrusão, baseado em software livre, utilizando o sistema operacional OpenBSD e o sistema de detecção de intrusão SNORT, visando reduzir o impacto causado por um ataque de Negação de Serviço. A metodologia consistiu em criar um ambiente virtual, onde foi possível simular uma rede e utilizando o ambiente criado, realizar um ataque de negação de serviços, permitindo que fosse possível avaliar o impacto causado pelo ataque e conseqüentemente, configurar o SNORT para detectá-lo. O objetivo principal, consistiu em alertar a equipe responsável pela rede, permitindo que estes possam tomar medidas para resolver o problema. Concluiu-se que, quando corretamente configurado, um sistema de detecção de intrusão torna-se um elemento fundamental em qualquer rede.

Palavras-chave: [Segurança da Informação. Negação de Serviço. Sistema de Detecção de Intrusão. SNORT. OpenBSD.]

ABSTRACT

This work presents a study about Intrusion Detection Systems, based on open source software, utilizing OpenBSD as the operational system and SNORT as the Intrusion Detection System, trying to reduce the impact caused by a Denial of Service attack. The methodology consisted on creating a virtual environment, where it was possible to simulate a network and, within the created environment, perform a Denial of Service attack, allowing the possibility of evaluating the impact caused by the attack and, consequently, configuring SNORT to detect it. The main objective was to alert the network responsible team, allowing that they could take measures to solve the problem. It was concluded that, when correctly configured, an Intrusion Detection System becomes a fundamental element in any network.

Key words: [Information Security. Denial of Service. Intrusion Detection System. SNORT. OpenBSD]

LISTA DE FIGURAS

<i>Figura 1: Estabelecimento da conexão TCP.....</i>	<i>21</i>
<i>Figura 2: Ataque por Reflexão</i>	<i>22</i>
<i>Figura 3: Ataque de Negação de Serviço Distribuído.</i>	<i>24</i>
<i>Figura 4: Modelo Básico de um IDS.....</i>	<i>27</i>
<i>Figura 5: Posicionamento do IDS.</i>	<i>31</i>
<i>Figura 6: Componentes do SNORT.</i>	<i>35</i>
<i>Figura 7: Ambiente de simulação do ataque.</i>	<i>42</i>
<i>Figura 8: Acessando o Site a partir da estação do atacante.....</i>	<i>44</i>
<i>Figura 9: Servidor Web Antes do Ataque - Saída do comando HTOP.....</i>	<i>45</i>
<i>Figura 10: Servidor Web Durante o Ataque - Saída do comando HTOP.....</i>	<i>46</i>
<i>Figura 11: Site totalmente inoperante durante o ataque.....</i>	<i>47</i>
<i>Figura 12: Tela principal do BASE - Ataque Detectado.....</i>	<i>48</i>
<i>Figura 13: Origem do Ataque.....</i>	<i>49</i>
<i>Figura 14: Origem do Ataque - Informações Detalhadas.....</i>	<i>49</i>
<i>Figura 15: Configurações do Firewall.....</i>	<i>55</i>
<i>Figura 16: Configuração dos Segmentos de Rede - Firewall.....</i>	<i>55</i>
<i>Figura 17: Criando os Segmentos de Rede.....</i>	<i>56</i>
<i>Figura 18: Visualizando o MAC Address.....</i>	<i>57</i>
<i>Figura 19: Saída do comando ifconfig.....</i>	<i>57</i>
<i>Figura 20: Saída do comando ifconfig - Configuração das interfaces.....</i>	<i>59</i>
<i>Figura 21: Início da Configuração do BASE.....</i>	<i>69</i>
<i>Figura 22: Configuração do caminho ADODB.....</i>	<i>69</i>
<i>Figura 23: Configuração do Banco de Dados.....</i>	<i>70</i>
<i>Figura 24: Configuração de usuário para administração.....</i>	<i>70</i>
<i>Figura 25: Criação das tabelas (BASE).....</i>	<i>71</i>
<i>Figura 26: Processo de criação das tabelas.....</i>	<i>71</i>
<i>Figura 27: Tela principal do BASE - Monitoramento.....</i>	<i>72</i>
<i>Figura 28: Propriedades da Máquina Virtual.....</i>	<i>76</i>
<i>Figura 29: Configuração da interface de rede.....</i>	<i>77</i>
<i>Figura 30: Configuração Estação do Atacante.....</i>	<i>78</i>
<i>Figura 31: Configuração da interface de rede.....</i>	<i>78</i>

SUMÁRIO

INTRODUÇÃO	9
PROBLEMA	10
JUSTIFICATIVA	10
OBJETIVOS	11
RESULTADOS A SEREM ALCANÇADOS	12
ORGANIZAÇÃO DO TRABALHO	12
1 CONCEITOS DE SEGURANÇA	14
1.1 Princípios da Segurança da Informação	15
1.2 Conceitos Relacionados	16
1.2.1 Vulnerabilidades	16
1.2.2 Ameaças	17
1.2.3 Incidente de Segurança	18
1.2.4 Riscos	18
1.2.5 Impacto	18
1.2.6 Controle	18
2 NEGAÇÃO DE SERVIÇOS	19
2.1 Conceitos	19
2.2 Tipos de Ataques	20
2.2.1 Consumo de largura de banda	20
2.2.2 Consumo de outros recursos	20
2.2.3 Ataque por Inundação (SYN Flood)	21
2.2.4 Ataque por reflexão	22
2.2.5 Ataques à Infraestrutura de Rede	23
2.2.6 Ataques por Vulnerabilidade	23
2.2.7 Ataque Distribuídos ou Coordenados (DDoS)	24
2.3 Medidas Preventivas	25
3 SISTEMA DE DETECÇÃO DE INTRUSÃO	27
3.1 Características	27
3.2 Tipos de IDS em Relação ao Posicionamento	28
3.2.1 Sistema de Detecção de Intrusão Baseado em Host (HIDS)	28
3.2.2 Sistema de Detecção de Intrusão Baseado em Rede (NIDS)	29
3.3 Localização do IDS na rede	31
4 OPENBSD e o SNORT	33
4.1 OpenBSD	33
4.1.1 Quem usa OpenBSD?	34
4.2 SNORT	35
4.2.1 Recursos do SNORT	35
5. AMBIENTE DE TESTES	38
5.1. Componentes	38
5.1.1 Firewall	38
5.1.2 DMZ (Zona Desmilitarizada) e Servidor Web	39
5.1.3 Rede Interna e Estação de Administração	40
5.1.4 Estação do Atacante (Simulação do Ataque)	40
5.1.5 Ferramenta de Ataque DoS	40
6 OPERACIONALIZAÇÃO DOS TESTES	42
6.1 Ambiente	42
6.2 Metodologia dos Testes	43
6.3 Resultados a Serem Alcançados	43
6.4 Execução dos Testes	44

6.5 Resultado dos Testes.....	45
CONCLUSÃO.....	51
REFERÊNCIAS	53
ANEXO I.....	55
A. IMPLANTAÇÃO E CONFIGURAÇÃO DO AMBIENTE VIRTUALIZADO	55
A.1 Configuração do Firewall.....	55
<i>A. 1.1 Configurando o Sistema Operacional.....</i>	<i>58</i>
<i>A. 1.2 Configurando as regras de filtragem.....</i>	<i>60</i>
A. 2 Instalação e configuração do SNORT.....	61
<i>A. 2.1 Instalação e configuração do MySQL-Server.....</i>	<i>62</i>
<i>A 2.2 Instalação e configuração do apache</i>	<i>63</i>
<i>A 2.3 Instalação e configuração do barnyard2.....</i>	<i>64</i>
<i>A 2.4 Instalação e configuração do BASE e suas dependências.....</i>	<i>66</i>
<i>A 2.5 Instalação e configuração do snort.....</i>	<i>73</i>
A 3 Configuração do servidor web.....	76
A 4 Configuração da estação do atacante.....	77
<i>A 4.1 Instalação do SLOWLORIS HTTP DOS.....</i>	<i>79</i>

INTRODUÇÃO

Nos dias atuais é quase impossível encontrar uma empresa que não esteja conectada à Internet. Seja apenas para permitir que seus funcionários a utilizem como uma forma de tornar o trabalho mais produtivo ou disponibilizando serviços para o meio externo, como um *WebSite*, um Serviço de *E-mail* entre outros. Estes serviços permitem que a empresa ultrapasse os limites de suas paredes, conseguindo alcançar uma quantidade de clientes muito maior e também permitir que seus funcionários consigam dar andamento a seus trabalhos sem necessariamente, estarem fisicamente na empresa.

Internet é sinônimo de informação ao alcance de milhares de pessoas ao mesmo tempo. No passado, quando se falava em uma empresa, logo vinha à mente um espaço físico como um prédio ou apenas um pequeno escritório, mas com a Internet essa visão vem mudando constantemente. Mais e mais empresas vêm expandindo seus negócios para a Internet, já que esta possibilita alcançar clientes que antes eram inalcançáveis, o que, conseqüentemente, gera um aumento considerável em seus lucros.

Hoje, qualquer pessoa que tenha um computador ou *smartphone* com acesso à internet, consegue pagar praticamente qualquer conta sem a necessidade de deslocar-se fisicamente até um caixa e ficar um bom tempo esperando na fila.

Com tantas facilidades, fica até difícil de se imaginar em ficar sem elas, mesmo que por algumas horas, o que infelizmente tem acontecido. Juntamente com toda a facilidade e comodidade que a internet proporciona, existem os perigos.

As empresas assim como os bancos vêm investindo pesado na segurança da informação, criando mecanismos que ofereçam mais segurança a seus clientes. A questão é que toda essa segurança para realizar uma transação eletrônica, de nada adiantará se o cliente nem sequer conseguir acessar a sua conta ou o site de vendas de sua preferência. É exatamente isso que um Ataque de Negação de Serviço (DoS) faz, tornar um determinado serviço indisponível.

Este estudo acadêmico tem o intuito de apresentar uma alternativa segura, que permita mitigar alguns Ataques de Negação de Serviço, utilizando para isso o Sistema de Detecção e Prevenção de Intrusão (IDS/IPS) chamado SNORT

(SNORT, 2013), um IDS/IPS baseado em software livre, bastante utilizado. Nesse projeto, o SNORT terá como base o Sistema Operacional OpenBSD (OPENBSD, 2013), um sistema operacional que tem a segurança como um de seus principais objetivos.

PROBLEMA

Para alguns problemas de segurança existentes, existe alguma tentativa de solução, como por exemplo os *anti-vírus*, como tentativa de solução para os *vírus*, os *anti-spams* como uma forma de reduzir a quantidade de mensagens indesejadas recebidas, porém, contra os ataques de negação de serviço, as ferramentas existentes não são totalmente eficazes (LAUFER et al., 2005). Mesmo havendo um empenho das áreas acadêmica e industrial, com o intuito de encontrar uma solução que consiga evitar e remediar esse tipo de ataque, as ferramentas existentes conseguem lidar apenas com ataques menos refinados, o que, consequentemente, fazem com que os ataques de negação de serviços sejam bem sucedidos na maioria dos casos (LAUFER et al., 2005).

Um ataque de negação de serviço, quando bem sucedido, deixa o serviço para o qual foi direcionado indisponível por tempo indeterminado, geralmente servidores *web* e *e-mail*, trazendo enormes prejuízos. Para um banco, ficar com seu sistema de *Internet Banking* parado é um prejuízo incalculável se levarmos em consideração não apenas os problemas financeiros, mas também o fato de que a imagem da instituição começa a ficar prejudicada, passando a sensação de insegurança para seus clientes.

Para um ataque tão poderoso como esse, já era de se esperar uma grande dificuldade de evitá-lo, sendo normal que surjam dúvidas como:

- Existe uma solução para se combater Ataques de Negação de Serviço?
- Como posso proteger uma infraestrutura de uma vulnerabilidade desse nível?

JUSTIFICATIVA

O SNORT (SNORT, 2013) foi escolhido não apenas pela vantagem de ser software livre, já que a questão de aquisição de licença é sempre uma dificuldade. O

SNORT também é um IDS já bastante conhecido e a sua configuração não é muito complicada.

O OpenBSD (OPENBSD, 2013) é conhecido pela ênfase dada a segurança, que é um de seus principais objetivos. De acordo com seu *website* oficial (OPENBSD, 2013), houve apenas duas vulnerabilidades remotas em sua instalação padrão.

O motivo dessa escolha é a criação de um ambiente leve, que permita um bom nível de segurança e com o mínimo de gastos, visando uma forma de mitigar os Ataques de Negação de Serviço.

OBJETIVOS

Objetivo Geral

O principal objetivo deste estudo acadêmico é tentar colocar em prática técnicas de detecção e prevenção de intrusão, voltadas principalmente para ataques de negação de serviço. Para tentar colocar essas técnicas em prática, será utilizado como base o Sistema Operacional OpenBSD (OPENBSD, 2013), cujos esforços estão direcionados a segurança (OPENBSD, 2013) e também o SNORT (SNORT, 2013), que é um Sistema de Detecção de Intrusão livre bastante utilizado.

Objetivos específicos

- Abordar os conceitos de segurança e sua importância;
- Apresentar o conceito de Negação de Serviço, abordando os tipos de ataques e medidas que podem ser tomadas;
- Configuração do OpenBSD;
- Instalação e Configuração do SNORT, tendo como foco a tentativa de mitigar a negação de serviços;
- Implementar uma rede (virtual) e simular um Ataque de Negação de Serviço;

- Controlar um ataque de negação de serviço e amenizar os impactos causados por ele.
- Ajustar as configurações do SNORT para que ele consiga alertar o administrador sobre um possível ataque de negação de serviço.

RESULTADOS A SEREM ALCANÇADOS

Espera-se que com esse estudo seja possível conseguir, de alguma forma, amenizar o impacto que um ataque de negação de serviço possa causar, simplesmente alertando o administrador da rede, para auxiliar na detecção do ataque.

Não existe ainda, até o presente momento, alguma técnica ou ferramenta que resolva definitivamente um ataque de negação de serviço e não é o foco desse estudo acadêmico apresentar uma solução definitiva, mas utilizar ferramentas de segurança, baseadas em software livre, contra esse tipo de ataque.

ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em seis capítulos e um anexo. O primeiro capítulo, aborda os conceitos de segurança da informação e seus princípios.

O capítulo 2 irá abordar os ataques de negação de serviço. Como é o seu funcionamento, seus conceitos, tipos de ataques e medidas preventivas.

No capítulo 3 serão abordados os sistemas de detecção de intrusão (IDS). O que são, como funcionam, quais suas características, quais os tipos de IDS e as vantagens e desvantagens de cada um e como posicionar corretamente um IDS em uma rede.

O capítulo 4 aborda brevemente os softwares utilizados: o OpenBSD e o SNORT.

O capítulo 5 irá descrever o ambiente de testes de forma geral, abordando toda a estrutura criada. Descrição do *Firewall*, servidor *Web*, estação responsável pela administração do IDS e a estação que irá simular o ataque de negação de serviço ao servidor *web*. Também irá descrever o objetivo da proposta apresentada.

No capítulo 6 será visto o ambiente de testes, mostrando os resultados de um ataque bem sucedido de negação de serviço e a importância de um sistema de detecção de intrusão implantado em uma rede.

No ANEXO 1, serão abordados a implantação e configuração do ambiente virtualizado. A configuração de cada ativo será descrita detalhadamente, assim como a instalação e configuração de cada serviço implantado.

1 CONCEITOS DE SEGURANÇA

A informação é um ativo que necessita ser adequadamente protegido, o que é de extrema importância em um ambiente cada vez mais interconectado. Devido ao aumento da interconectividade, a informação está agora exposta a um crescente número e a uma grande variedade de ameaças e vulnerabilidades. A informação existe em diferentes formas, podendo ser impressa ou escrita em papel, armazenada eletronicamente ou apenas falada. Não importa como a informação está sendo apresentada ou o meio pelo qual ela é compartilhada ou armazenada, o que importa é que ela seja protegida da forma mais adequada possível (ABNT, 2005).

De acordo com Campos (2006), informação é um elemento essencial para a geração do conhecimento para a tomada de decisões e representa valor para o negócio, dentro de cada um de seus processos.

A segurança da informação é a proteção da informação contra vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio e é obtida a partir da implementação de um conjunto de controles adequados, incluindo políticas, processos, procedimentos, estruturas organizacionais e funções de software e hardware (ABNT, 2005).

Ao mesmo tempo em que as informações são consideradas os principais patrimônios de uma organização, estão também sob risco constante, talvez como nunca antes estiveram. Isso fez com que a segurança da informação se tornasse um ponto crucial para a sobrevivência das organizações. Quando as informações eram armazenadas em papel, pode-se dizer que a segurança era muito mais simples, contudo, devido as mudanças tecnológicas, a estrutura de segurança ficou mais sofisticada. Com o crescimento das redes que conectam o mundo inteiro, os aspectos de segurança atingiram um nível de complexidade muito maior, ao ponto de haver a necessidade do desenvolvimento de equipes cada vez mais especializadas para sua implementação e gerenciamento (FERREIRA, 2003).

As informações contidas em sistemas informatizados são consideradas como recursos críticos para a grande maioria das empresas e são de suma impor-

tância para a concretização de negócios e tomada de decisões. Para essas empresas, uma falha de segurança pode ocasionar prejuízos incalculáveis. Imagine o que poderia acontecer se as informações de determinada empresa caírem nas mãos da concorrência, se forem copiadas, totalmente apagadas ou se não puderem ser acessadas para o fechamento de um grande negócio. Atacar esses sistemas não é algo muito complexo, já que os sistemas de informação podem estar conectados a redes externas (FERREIRA, 2003).

Confidencialidade, integridade e disponibilidade da informação podem ser essenciais para preservar a competitividade e principalmente a imagem da empresa perante o mercado e os clientes. Os sistemas de informação e as redes de computadores dessas organizações estão sendo colocados à prova por uma enorme diversidade de ameaças, desde fraudes eletrônicas, espionagem, sabotagem, vandalismo e inundação. Problemas causados por vírus, *hackers* e ataques de negação de serviço estão se tornando comuns e cada vez mais sofisticados (FERREIRA, 2003).

As organizações estão cada vez mais vulneráveis às ameaças de segurança, pois estão cada vez mais dependentes dos sistemas de informação. A conexão de redes públicas e privadas e também o compartilhamento de recursos, acabam dificultando a implementação de um controle de acesso centralizado que seja realmente eficiente (FERREIRA, 2003).

Pode-se perceber naturalmente que a informação é um ativo da organização, um bem que deve ser tão protegido quanto os bens físicos, tendo em vista a sua importância para a própria existência da organização (CAMPOS, 2006).

1.1 Princípios da Segurança da Informação

Os princípios básicos da segurança da informação são a Confidencialidade, a Integridade e a Disponibilidade. Caso um desses princípios seja desrespeitado em algum momento, isto significa uma quebra de segurança da informação, o que também pode ser chamado de incidente de segurança da informação (CAMPOS, 2006).

Confidencialidade – Toda informação deve ser protegida de acordo com o grau de sigilo de seu conteúdo, visando a limitação de seu acesso e uso apenas às pessoas para quem elas são destinadas (SÊMOLA, 2003). Este princípio é respeitado, quando apenas as pessoas explicitamente autorizadas podem ter acesso à informação (CAMPOS, 2006). Quando uma informação é acessada por pessoa não-autorizada, seja de forma intencional ou não, então isto é caracterizado com sendo um incidente de segurança da informação devido a quebra de confidencialidade (CAMPOS, 2006).

Integridade – Toda informação deve ser mantida na mesma condição em que foi disponibilizada pelo seu proprietário, visando protegê-las contra alterações indevidas, intencionais ou acidentais (SÊMOLA, 2003). Este princípio é respeitado quando a informação acessada está completa, sem alterações, tornando-a confiável (CAMPOS, 2006). Caso uma informação seja alterada indevidamente, intencionalmente ou não, então houve o que chamamos de um incidente de segurança da informação por quebra de integridade (CAMPOS, 2006).

Disponibilidade – Toda informação gerada ou adquirida por um indivíduo ou instituição deve estar disponível aos seus usuários no momento em que os mesmos necessitarem (SÊMOLA, 2003). Este princípio é respeitado quando a informação está acessível, por pessoas autorizadas, sempre que necessário (CAMPOS, 2006). Quando a informação não pode ser acessada nem mesmo por quem é de direito, como por exemplo no caso em que um sistema esteja indisponível, seja pelo fato de estar sendo realizada uma manutenção ou o servidor tiver sido vítima de uma ataque de negação de serviço, então isto é um incidente de segurança da informação por quebra de disponibilidade (CAMPOS, 2006).

1.2 Conceitos Relacionados

1.2.1 Vulnerabilidades

Vulnerabilidades são fraquezas presentes nos ativos de informação, que podem causar de forma intencional ou não, a quebra de um ou mais dos três pilares da segurança da informação. As vulnerabilidades podem ser relacionadas aos seguintes ativos de informação citadas em alguns exemplos abaixo (CAMPOS, 2006):

Tecnologias: podem ser computadores sem nenhum tipo de proteção contra vírus ou uma outra ameaça; dispositivos de impressão e fax localizados em locais de acesso público; rede local acessível por senha padrão ou pública; acesso a áreas críticas de informação sem controle de acesso físico.

Pessoas e processos: ausência de uma política institucional de segurança dentro da organização; colaboradores não-treinados e conseqüentemente desconhecedores da política de segurança da informação da organização; ausência de procedimentos disciplinares para o tratamento das violações da política de segurança.

Ambiente: ausência de mecanismos contra incêndio; ausência de mecanismos de prevenção a enchentes;

Essas vulnerabilidades poderão ou não serem exploradas e é possível que um ativo possua uma vulnerabilidade que nunca será explorada (CAMPOS, 2006).

As vulnerabilidades por si só não provocam incidentes, pois são elementos passivos, necessitando de um agente causador ou condição favorável, que são as ameaças (SÊMOLA, 2003).

1.2.2 Ameaças

Pode-se definir ameaça como um agente externo ao ativo de informação, que poderá de alguma forma aproveitar-se das vulnerabilidades desse ativo, acarretando na quebra de um dos princípios da segurança da informação. Diminuir ou mesmo acabar com as ameaças é algo muito difícil, simplesmente pelo fato de serem imprevisíveis e também por estarem fora do nosso controle. (CAMPOS, 2006).

As ameaças podem ser classificadas quanto a sua intencionalidade, podendo ser divididas nos seguintes grupos (SÊMOLA, 2003).

- **Naturais** – são ameaças decorrentes de fenômenos da natureza.
- **Involuntárias** – são ameaças inconscientes, quase sempre causadas pelo desconhecimento.

- **Voluntárias** – ameaças propositais causadas por agentes humanos, como por exemplo hackers, invasores, criadores de vírus.

1.2.3 *Incidente de Segurança*

Um incidente de segurança da informação é quando uma ameaça explora as vulnerabilidades de um ativo de informação, afetando de alguma forma o negócio da organização (CAMPOS, 2006).

É um evento decorrente da ação de uma ameaça, que explora uma ou mais vulnerabilidades, levando a perda de princípios da segurança da informação. Um incidente gera impactos aos processos de negócios da empresa, sendo ele o elemento a ser evitado em uma cadeia de gestão de processos e pessoas (SÊMOLA, 2003).

1.2.4 *Riscos*

É a chance de uma falha de segurança ocorrer levando-se em conta o grau das vulnerabilidades presentes nos ativos e o grau das ameaças que possam explorar essas vulnerabilidades. Um ponto importante a se observar é com relação ao grau de impacto que um incidente de segurança poderia causar se ocorresse com um determinado ativo de informação (CAMPOS, 2006).

Probabilidade de ameaças explorarem vulnerabilidades, acarretando em possíveis impactos aos negócios (SÊMOLA, 2003).

1.2.5 *Impacto*

O impacto de um incidente são as potenciais consequências que este incidente possa causar ao negócio da organização (CAMPOS, 2006).

Extensão dos danos causados por um incidente de segurança sobre um ou mais processos de negócio (SÊMOLA, 2003).

1.2.6 *Controle*

É todo e qualquer mecanismo utilizado para diminuir a fraqueza ou a vulnerabilidade de um ativo, seja um equipamento ou tecnologia, uma pessoa ou processo (CAMPOS, 2006).

2 NEGAÇÃO DE SERVIÇOS

Basicamente, os Ataques de Negação de Serviços são ataques cuja principal característica, seja a de que o atacante tenta de alguma forma, impedir que usuários legítimos acessem determinado serviço (CERT, 1997), tendo como principais objetivos os citados abaixo:

- Inundação da rede, impedindo todo e qualquer tráfego de rede legítimo.
- Interrupção de conexões entre duas máquinas, impedindo dessa forma o acesso a algum serviço.
- Tentativa de impedir que um indivíduo consiga acessar determinado serviço.
- Tentativa de interromper ou negar um serviço a uma pessoa ou sistema específico.

Não se pode caracterizar todas as interrupções de serviços, mesmos aquelas que resultam de atividades maliciosas como sendo um ataque de negação de serviços. Outros tipos de ataques podem incluir uma negação de serviço como um componente de um ataque ainda maior (CERT, 1997).

2.1 Conceitos

Em um ataque de negação de serviço, o atacante explora a conectividade com a internet para tentar paralisar os serviços oferecidos por um site vítima, na maioria das vezes simplesmente inundando a vítima com várias solicitações. Um ataque de negação de serviço, pode ser tanto um ataque de fonte única, ou seja, que é originária de um único *host*, ou múltiplas fontes, onde nesse caso, vários *hosts* são coordenados para inundar a vítima com uma enxurrada de pacotes. Este último ataque citado é chamado de Ataque de Negação de Serviço Distribuído (DDoS) (HUSSAIN; HEIDEMANN; PAPADOPOULOS, 2003).

Desenvolver técnicas para detectar e responder a ataques rapidamente é algo essencial, pois com o passar dos anos, ataques de negação de serviço tem

causados danos financeiros significativos (HUSSAIN; HEIDEMANN; PAPADOPOULOS, 2003).

2.2 Tipos de Ataques

Ataques de negação de serviços vêm em uma variedade de formas, visando uma variedade de serviços. Alguns tipos de ataque objetivam (CERT, 1997):

- consumo de recursos escassos, limitados ou não renovável
- destruição ou alteração de informações de configuração
- destruição ou alteração física de componentes de rede

2.2.1 Consumo de largura de banda

Um atacante pode ser capaz de consumir toda a largura de banda disponível na rede, gerando uma grande quantidade de pacotes direcionados a ela. Geralmente esses pacotes são pacotes ICMP *ECHO*, mas nada impede que seja qualquer coisa. O atacante pode utilizar várias máquinas para coordenar o ataque em diferentes redes (CERT, 1997).

2.2.2 Consumo de outros recursos

O atacante pode ser capaz de consumir outros recursos necessários para o funcionamento do sistema. Em muitos sistemas, uma quantidade limitada de estruturas de dados estão disponíveis para armazenar informações de processos e um invasor poderia ser capaz de consumir essas estruturas de dados, simplesmente escrevendo um *script* que não faz absolutamente nada, mas que repetidamente cria cópias de si mesmo, o que poderia preencher toda a tabela de processos consumindo toda a CPU (CERT, 1997).

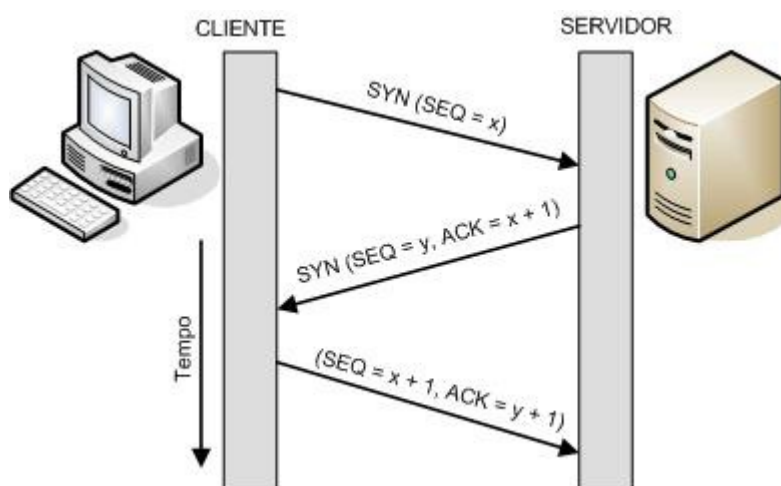
O espaço em disco também poderia ser todo consumido. O atacante poderia gerar um quantidade excessiva de mensagens de correio, ou qualquer outra ação que permita que dados sejam gravados no disco, até conseguir executar uma negação de serviço (CERT, 1997).

2.2.3 Ataque por Inundação (SYN Flood)

Um dos tipos de ataques mais frequentes é o que é executando contra a conectividade de rede, cujo objetivo é evitar a comunicação de máquinas com a rede. Um bom exemplo para este tipo de ataque e também um dos mais conhecidos é *SYN flood* (inundação de SYN) (CERT, 1997).

O ataque por inundação de segmentos TCP SYN, explora o procedimento de abertura de conexão do protocolo TCP (LAUFER et al., 2005). A conexão TCP inicia-se com a negociação de alguns parâmetros entre o cliente e o servidor, conforme pode ser visto na Figura 1. O cliente envia um segmento TCP SYN para o servidor indicando uma solicitação de abertura de conexão, onde, nesse segmento, contém um parâmetro denominado número de sequência inicial. Esse número de sequência permite que o receptor reconheça dados perdidos, repetidos ou que estejam fora de ordem. Assim que o segmento TCP SYN é recebido o servidor precisa de tempo para processar o pedido de conexão e então poder alocar memória para armazenar informações sobre o cliente. Logo após, um segmento TCP SYN/ACK é enviado como resposta, notificando o cliente que seu pedido de conexão foi então aceito. O número de sequência do cliente é reconhecido pelo segmento de resposta, que então envia o número de sequência inicial do servidor. Para finalizar, o cliente envia um segmento TCP ACK para reconhecer o número de sequência do servidor e poder então completar a abertura da conexão (LAUFER et al., 2005).

Figura 1: Estabelecimento da conexão TCP



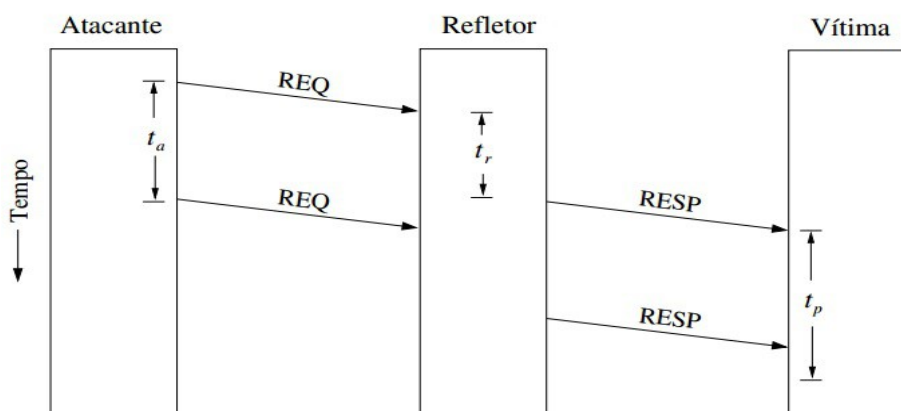
Fonte: próprio autor, baseado em TANENBAUM; WETHERALL (2011, p.352)

Neste tipo de ataque, o atacante inicia o processo de estabelecimento de uma conexão para máquina da vítima, mas faz de uma forma que impede a finalização da conexão. A máquina vítima, tem reservado apenas um número limitado de estruturas de dados, necessárias para completar a conexão e como resultado, as conexões legítimas serão negadas, enquanto a máquina da vítima estiver esperando para completar as falsas conexões “semi abertas” (CERT, 1997).

2.2.4 Ataque por reflexão

O ataque por reflexão é também um ataque por inundação, que tem como objetivo consumir os recursos da vítima. Na Figura 2, pode-se ver como ocorre o ataque por reflexão, utilizando uma estação intermediária entre o atacante e a vítima, onde a estação intermediária é utilizada para refletir o tráfego de ataque em direção à vítima. Essa ação torna ainda mais difícil a descoberta da identidade dos atacantes, pois o tráfego que chega até a vítima, é originado no refletor, ao invés de ser no próprio atacante. O atacante precisa enviar algum tipo de requisição (REQ) para o refletor, utilizando o endereço da vítima como sendo o endereço de origem, ao invés de utilizar o seu próprio endereço. Quando o refletor receber uma requisição, ele não irá conseguir verificar a autenticidade da origem da requisição e então enviará um resposta (RESP) diretamente para a vítima (LAUFER et al., 2005).

Figura 2: Ataque por Reflexão



Fonte: LAUFER et al (2005, p.22)

Neste ataque, a máquina utilizada como refletor, pode ajudar a consumir os recursos da vítima, desde que a mensagem de requisição enviada pelo atacante

seja menor que a mensagem de resposta enviada pelo refletor, fazendo com que o refletor opere como um amplificador do tráfego de ataque (LAUFER et al., 2005).

2.2.5 Ataques à Infraestrutura de Rede

Ataques direcionados a uma infraestrutura de redes, como um grande *website* ou serviços de resolução de nomes (DNS) de grandes organizações, precisam ser realizados em larga escala, para que se consiga obter sucesso, pois geralmente a infraestrutura dessas vítimas são superdimensionadas, com bastante memória e processamento. Neste ataque, o atacante poderia tentar, como alternativa, consumir toda a banda da vítima utilizando o tráfego de ataque, pois mesmo que os servidores tenham poder de processamento e memória para conseguir atender a todas as requisições que chegam, várias dessas requisições ainda seriam perdidas na infraestrutura de rede, como por exemplo em algum roteador entre o atacante e a vítima, já que o tráfego direcionado à vítima poderá ser maior do que o enlace de saída poderia suportar (LAUFER et al., 2005).

2.2.6 Ataques por Vulnerabilidade

Deixar os serviços da vítima inoperantes também é uma forma de negação de serviço. Este objetivo pode ser alcançado explorando alguma vulnerabilidade na implementação da pilha de protocolos ou em alguma aplicação da própria vítima, deixando-a inoperante por um longo tempo (LAUFER et al., 2005).

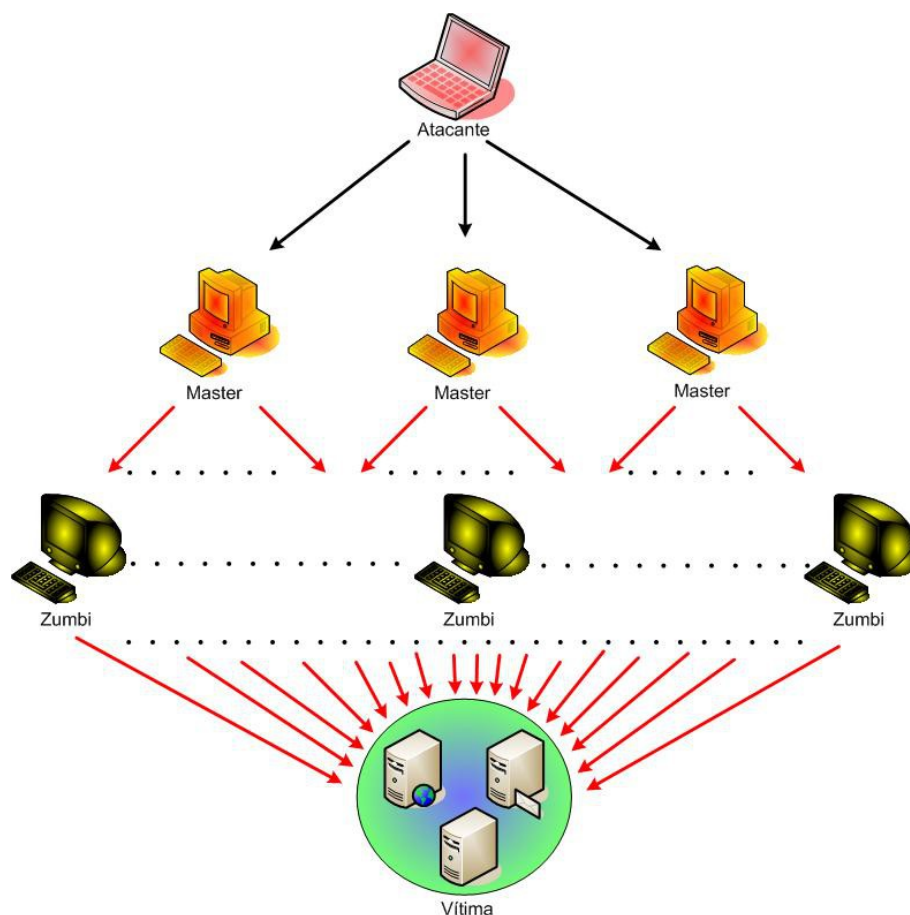
Vulnerabilidades no protocolo IP também já foram exploradas, tirando-se proveito da capacidade do protocolo de permitir a quebra do pacote em pacotes menores, quando este é muito grande para ser transmitido sobre determinada tecnologia de rede, enviando cada fragmento deste pacote separadamente. Devido a essa fragmentação, cada fragmento recebe um identificador, fazendo com que o receptor consiga juntar os fragmentos de um mesmo pacote e recebe também um número de sequência, para que seja possível determinar em que parte, no pacote original, determinado fragmento se encontra. Dessa forma, o receptor então poderá receber os fragmentos e concatená-los, conseguindo assim, remontar o pacote original. Partindo desse princípio, o objetivo do ataque consistia no envio de vários fragmentos IP, que pertenciam ao mesmo pacote com números de sequência que se sobrepunham, fazendo com que a vítima recebesse alguns poucos fragmentos intencionalmente

mal formados. Por esse motivo, o protocolo entrava em um estado não previsto, causando o seu congelamento e até mesmo a sua reinicialização (LAUFER et al., 2005).

2.2.7 Ataque Distribuídos ou Coordenados (DDoS)

Os ataques coordenados, também conhecidos como Ataques de Negação de Serviço Distribuídos (DDoS) é uma evolução do Ataque de negação de Serviço (NAKAMURA; GEUS, 2007). Esse ataque tem como objetivo, fazer com que diversos *hosts* distribuídos sejam atacados e coordenados pelo atacante, com o intuito de realizar ataques simultâneos aos alvos, resultando em um ataque bastante eficiente, onde a vítima fica quase que totalmente indefesa e sem a possibilidade de conseguir identificar a origem dos ataques, já que eles se originam de *hosts* intermediários controlados pelo atacante (NAKAMURA; GEUS, 2007).

Figura 3: Ataque de Negação de Serviço Distribuído.



Fonte: próprio Autor. Baseado em NAKAMURA; GEUS (2007 p.116)

Na Figura 3 podemos ver o funcionamento de um ataque de negação de serviço distribuído, onde o atacante define os sistemas *master*, que se comunicam com as máquinas Zumbis. Ambos os tipos de máquinas, foram de alguma forma, comprometidas pelo atacante, permitindo que estas sejam controladas remotamente. São essas máquinas zumbis que realizam o ataque à vítima. Assim como os zumbis, as máquinas *master* também são vítimas do atacante, que conseguiu explorar alguma vulnerabilidade conhecida, permitindo a instalação de processos utilizados durante o ataque (NAKAMURA; GEUS, 2007).

As ferramentas de ataque DDoS, tiram proveito das melhores tecnologias de ataque existentes, como por exemplo, o uso de criptografia durante a comunicação entre o atacante e os *hosts masters* e zumbis (NAKAMURA; GEUS, 2007). Uma infinidade dessas ferramentas, cada vez mais sofisticadas, para realizar um ataque de negação de serviço e que automatizam o processo de comprometer as vítimas, estão disponíveis na Internet, com instruções detalhadas, permitindo que até mesmo um amador consiga utilizá-las de forma eficiente (HUSSAIN; HEIDEMANN; PAPA-DOPOULOS, 2003).

2.3 Medidas Preventivas

Ataques de Negação de Serviços podem resultar em enormes perdas financeiras, então é de suma importância que as organizações se atentem a esse problema. De acordo com o CERT (1997), algumas ações podem ser realizadas para se tentar evitar ou diminuir o impacto de um ataque de negação de serviço. Vejamos algumas recomendações:

- Implementação de filtros nos roteadores, reduzindo sua exposição a certos ataques de negação de serviços, como *SYN Flood* por exemplo.
- Manter o sistema sempre atualizado. Sempre que estiverem disponíveis, instalar os *patches* de segurança.
- Desativar todos os serviços de rede que não são utilizados ou desnecessários, limitando dessa forma, a capacidade de um intruso conseguir explorar alguma vulnerabilidade desses serviços e conseguir executar um ataque.

- Monitorar o desempenho do sistema, estabelecendo linhas de base para se definir o que é atividade normal e conseguir dessa forma, detectar níveis inco-muns de atividade de disco, uso de CPU ou tráfego de rede.
- Utilizar alguma ferramenta que consiga detectar alterações em arquivos de configuração ou qualquer outro arquivo vital para o funcionamento do siste-ma.

Como podemos ver, manter-se totalmente protegido de uma ataque de negação de serviços é algo extremamente difícil. Existem diferentes formas de se deixar uma vítima impossibilitada de fornecer seus serviços. Por mais que a organi-zação mantenha seus sistemas atualizados e seus servidores possuam uma grande quantidade de recursos disponíveis, ainda assim seria possível realizar um ataque, tirando proveito da infraestrutura de rede.

É importante ressaltar, que, quando não é possível evitar o ataque, ainda é possível tentar tomar medidas reativas, onde pode-se tentar por exemplo, detectar a origem do ataque, o que também não é algo tão simples quanto parece, pois devi-do à arquitetura da Internet, os endereços podem ser forjados, impossibilitando des-sa forma, de se confiar totalmente no endereço de origem do pacote (LAUFER et al., 2005).

3 SISTEMA DE DETECÇÃO DE INTRUSÃO

Um sistema de detecção de intrusão (IDS), é considerado um componente básico, em qualquer ambiente que se preocupe com a segurança de sua estrutura de rede como um todo. Possui a capacidade de detectar uma grande variedade de ataques e intrusões, auxiliando na proteção (NAKAMURA; GEUS, 2007).

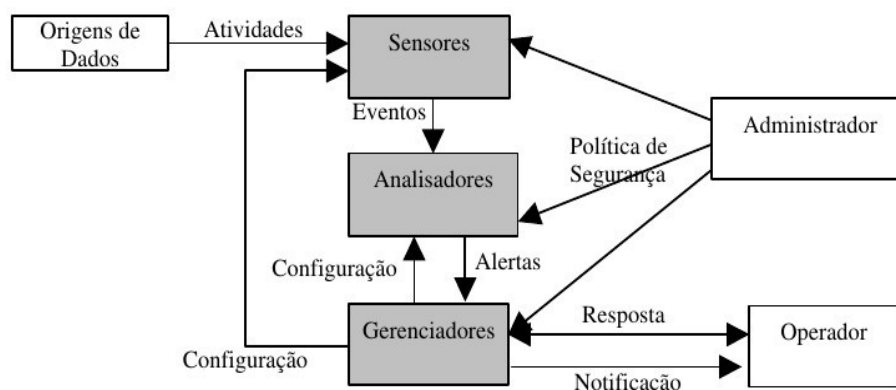
Um IDS, consegue detectar alterações, manipulações indesejadas em sistemas de computadores, geralmente vindas da Internet. É uma ferramenta utilizada para detectar uma grande quantidade de ataques ou comportamentos maliciosos, que podem de alguma forma, comprometer a segurança e a confiabilidade de um sistema (LEOBONS, 2007).

3.1 Características

Podemos assimilar um IDS com uma câmera, semelhante aquelas de circuito interno, que fica registrando tudo o que acontece, ou a um alarme contra acesso não autorizado, podendo realizar a detecção baseado em algum tipo de conhecimento, como por exemplo, assinaturas de ataques, ou em mudanças de comportamentos. Quando os primeiros sinais de um possível ataque é detectado, de forma lógica e compreensível, os perigos de um ataque real podem ser minimizados (NAKAMURA; GEUS, 2007).

Além de detectar, o IDS também é capaz de alertar os administradores quanto a possíveis intrusões. Informações referentes as tentativas de ataques, podem ser conseguidas através desses sistemas. Um IDS funciona combinando uma série de funções, que por trabalharem de modo integrado, consegue detectar, analisar e responder a atividades consideradas suspeitas (NAKAMURA; GEUS, 2007).

Figura 4: Modelo Básico de um IDS



A Figura 4 ilustra um modelo básico de um IDS, onde é possível entender o relacionamento entre os elementos de um sistema de detecção de intrusão, formado por:

Origens de dados, que é onde as atividades de sujeitos sobre objetos acontecem; Os **Sensores** são responsáveis por gerar os eventos, onde os **Analísadores** poderão verificar indícios de anomalias, podendo então, gerar os **Alertas**. Responsável pela configuração do sistema, temos os **Gerenciadores**, que a partir dos alertas, se encarregam de enviar as notificações aos operadores. Existe também o fator humano, na função de **Administrador** do sistema, definindo as políticas de segurança. A existência de mecanismos de **resposta** auxilia o rastreamento de eventos suspeitos, permitindo identificar e responsabilizar corretamente o sujeito atacante (BRANDÃO, 2007).

3.2 Tipos de IDS em Relação ao Posicionamento

Os principais tipos de IDS, são os baseados em *host* (*Host-Based Intrusion Detection System* – HIDS) e os baseados em rede (*Network-Based Intrusion Detection System* - NIDS). Aproveitando as melhores características de ambos, acabaram surgindo também os IDS Híbridos (*Hybrid IDS*) (NAKAMURA; GEUS, 2007).

3.2.1 Sistema de Detecção de Intrusão Baseado em Host (HIDS)

Foram os primeiros tipos de IDS a serem desenvolvidos. Esses sistemas, conseguem coletar e analisar os dados originados nos computadores que hospedam um determinado serviço, como por exemplo, um servidor de e-mail ou web. Esses dados podem ser analisados localmente, ou então serem enviados para um servidor central onde poderão ser analisados (LEOBONS, 2007).

O HIDS realiza o monitoramento do sistema, tomando como base informações de arquivos de *logs* ou de agentes de auditoria. Também é capaz de monitorar acessos e alterações em arquivos importantes do sistema, alterações nos privilégios dos usuários, processos do sistema, utilização da CPU, entre outras possíveis anomalias. Por meio de *checksums*, o HIDS consegue checar a integridade dos arquivos do sistema. Tal característica é de grande importância, pois arquivos corrompidos, que poderiam ser algum tipo de *backdoors*, são detectados antes que consi-

gam causar alguma problema. Por na maioria da vezes, os HIDS não conseguirem emitir alertas em tempo real, estes são considerados em alguns momentos apenas como ferramentas, ao invés de sistemas (NAKAMURA; GEUS, 2007).

Algumas vantagens do HIDS (NAKAMURA; GEUS, 2007):

- Podem verificar o sucesso ou falha de um ataque, tendo como base os *logs* do sistema;
- Ataques que tiram proveito da criptografia podem não ser detectados pela rede, mas conseguem ser detectados pelo HIDS, devido ao fato do sistema operacional decifrar os pacotes que chegam ao equipamento;
- Não depende da topologia de rede, o que permite que seja utilizado em redes separadas por *switches*.

Algumas desvantagens do HIDS (NAKAMURA; GEUS, 2007):

- O gerenciamento e configuração em todos os *hosts*, torna-se algo difícil, causando problemas de escalabilidade;
- É dependente do sistema operacional;
- Pode haver perda de informação caso o HIDS seja invadido;
- O *host* monitorado pode apresentar uma queda de desempenho.

3.2.2 Sistema de Detecção de Intrusão Baseado em Rede (NIDS)

Geralmente com a interface de rede atuando em modo *promíscuo*, o NIDS monitora todo o tráfego de rede. A detecção dá-se com a captura e análise dos cabeçalhos e conteúdos dos pacotes, onde serão comparados com padrões ou assinaturas conhecidas. A eficiência do NIDS, mostra-se principalmente contra ataques como *port scanning*, *IP spoofing* ou *SYN flooding*. Também consegue detectar ataques de *buffer overflow*. Pode-se dividir o NIDS em duas partes que trabalham em conjunto, sendo estas os sensores e o gerenciador (NAKAMURA; GEUS, 2007).

Os sensores ficam espalhados pela rede, são estes os responsáveis pela captura, formatação de dados e análise do tráfego da rede. Já o gerenciador é o res-

ponsável por fazer com que os sensores sejam administrados de forma integrada, contendo a definição dos tipos de resposta a serem utilizados, para cada tipo de comportamento detectado como suspeito (NAKAMURA; GEUS, 2007).

Ao contrário do HIDS, que analisam as informações que residem e são originadas em um computador, o NIDS utiliza técnicas como *packet-sniffing*, capturando dados da pilha TCP/IP ou pacotes de outros protocolos passando pela rede (LEOBONS, 2007).

A capacidade de detectar ataques na rede em tempo real, é uma característica de grande importância do NIDS. Com o sensor atuando em modo promíscuo, no mesmo segmento de rede de um servidor atacado, ele consegue capturar os pacotes referentes ao ataque, analisá-lo e responder ao ataque quase que instantaneamente (NAKAMURA; GEUS, 2007).

Algumas vantagens do NIDS (NAKAMURA; GEUS, 2007):

- O monitoramento pode ser fornecido para diferentes plataformas;
- O NIDS pode monitorar atividades suspeitas em portas conhecidas;
- Os ataques podem ser detectados e identificados em tempo real, permitindo ao usuário determinar rapidamente o tipo de resposta mais apropriado;
- Detecta não apenas os ataques, mas também as tentativas de ataques que não tiveram sucesso.
- Não causa impacto no desempenho da rede.

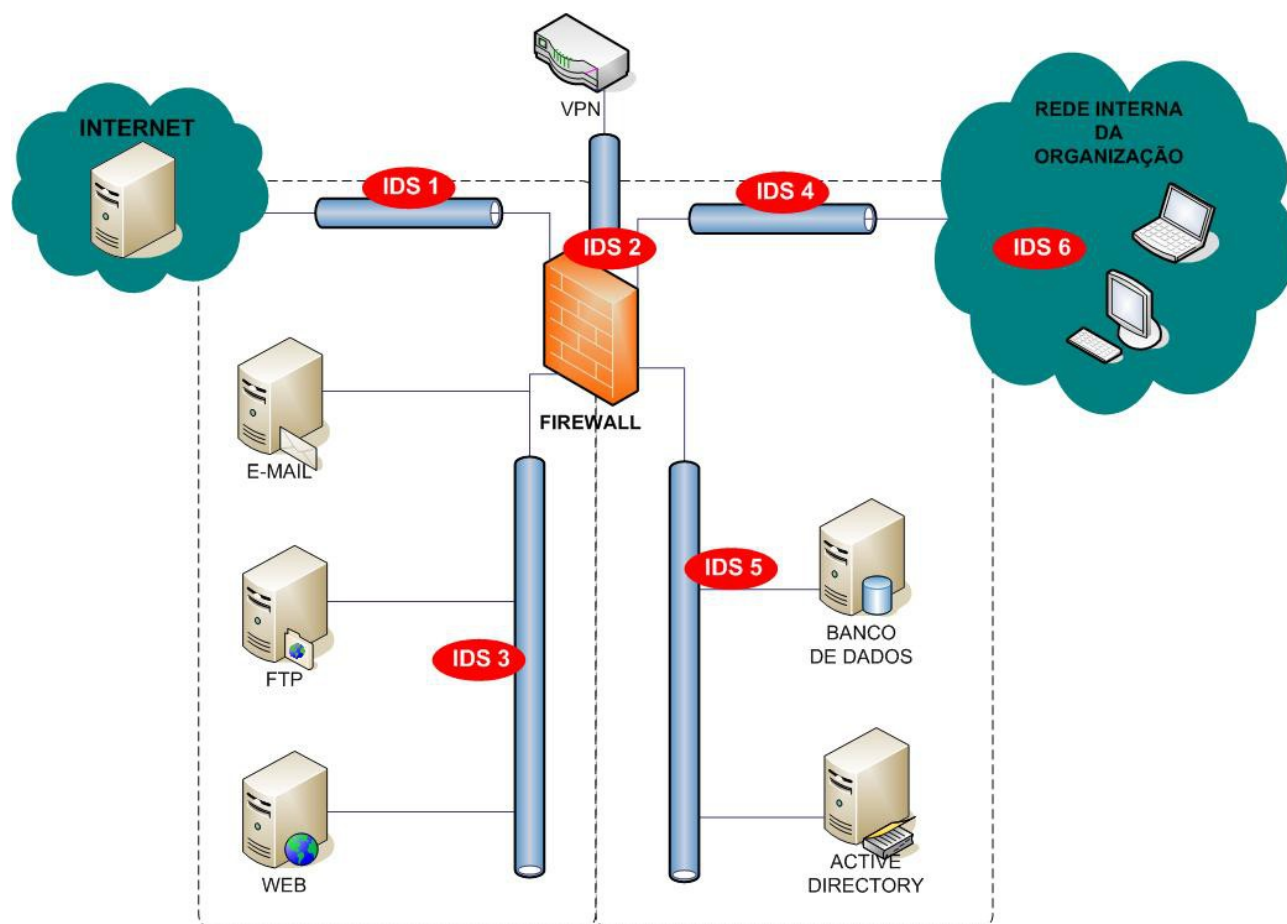
Algumas desvantagens do NIDS (NAKAMURA; GEUS, 2007):

- Em redes saturadas pode haver perdas de pacotes;
- Não consegue monitorar tráfego cifrado;
- Sua utilização em redes segmentadas é difícil.

3.3 Localização do IDS na rede

Pode-se posicionar o IDS em vários lugares na rede, já que cada posição visa um tipo específico de proteção. Na Figura 5 podemos observar algumas das posições em que o IDS pode ser configurado (NAKAMURA; GEUS, 2007).

Figura 5: Posicionamento do IDS.



Fonte: Próprio autor. Baseado em: NAKAMURA; GEUS (2007. p.296)

- **IDS 1:** todas as tentativas de ataque contra a rede são detectadas, inclusive as tentativas que não teriam nenhum efeito. Essa localização fornece uma grande quantidade de informações quanto aos tipos de tentativas de ataques.
- **IDS 2:** configurado no próprio *firewall*. Nesse cenário o IDS pode detectar as tentativas de ataque contra o *firewall*.

- **IDS 3:** tentativas de ataque contra os servidores da DMZ que são capazes de passar pelo *firewall*, são detectados pelo IDS.
- **IDS 4:** detecta tentativas de ataque contra a rede interna da organização, que conseguiram passar pelo *firewall* e que podem passar pela VPN.
- **IDS 5:** detecta tentativas de ataque contra servidores na DMZ 2, que passaram pelo *firewall*, pela VPN ou algum serviço proveniente da DMZ 1.
- **IDS 6:** tentativas de ataques internos são detectados.

O IDS posicionado antes do *firewall*, como por exemplo no caso do IDS 1, a detecção ocorre simultaneamente ao ataque. Quando o IDS fica depois do *firewall*, a detecção passa a ser de intrusões, já que o atacante, nesse caso, já passou pelo *firewall* (NAKAMURA; GEUS, 2007).

Neste trabalho abordaremos o modelo de IDS 2, configurado no próprio *Firewall*.

4 OPENBSD e o SNORT

4.1 OpenBSD

Souza (2009), define o OpenBSD como o melhor sistema operacional BSD voltado para segurança de dados. Todo o projeto do OpenBSD foi desenvolvido por profissionais voltados para a área de segurança, fazendo um sistema operacional com criptografia de *swap* e auditoria de arquivos por padrão, automaticamente logo após a instalação.

O projeto OpenBSD, produz um sistema operacional livre, cujos esforços enfatizam a portabilidade, padronização, correção, segurança pró-ativa e criptografia integrada. Possuir uma segurança extremamente forte é uma das metas do projeto, que aspira ser o número um na indústria de segurança. Diferentemente de muitos vendedores de software, o projeto acredita na divulgação completa de todos os problemas de segurança encontrados (OPENBSD, 2013).

Possui uma equipe de auditoria de segurança, que procura continuamente corrigir novas falhas de segurança desde 1996. Durante o processo de auditoria, uma análise é realizada, abrangendo arquivo por arquivo de cada componente de software crítico. Durante o processo de auditoria, muitos erros podem ser encontrados e mesmo que uma possível exploração decorrente do erro não esteja provada, este é corrigido assim mesmo e a equipe segue em frente na busca de novos erros (OPENBSD, 2013).

O sistema operacional vem em um modo batizado de seguro por padrão, onde todos os serviços não-essenciais estão desativados e conforme o usuário ou o administrador vai se tornando mais familiarizado com o sistema, ele vai aos poucos habilitando o que é realmente necessário para atender a sua necessidade, diferentemente de outros sistemas operacionais, que após a instalação, já estão com vários serviços habilitados, criando instantaneamente sérios problemas de segurança para os usuários dentro de minutos (OPENBSD, 2013).

4.1.1 Quem usa OpenBSD?

OpenBSD é utilizado em vários países ao redor do mundo, desde universidades, órgãos públicos e empresas privadas. Podemos ver alguns exemplos abaixo (OPENBSD, 2013):

Human Rights and Equal Opportunity Commission, Australia: Sediada em Sydney, é uma organização independente que administra as leis federais relativas as violações dos direitos humanos e discriminação. OpenBSD está sendo utilizado para oferecer vários serviços de rede.

Ministério de Obras Públicas do Governo do Chile: possui uma WAN nacional e o OpenBSD é utilizado em seus *firewalls* e balanceadores de *link*. O sistema é utilizado desde 2001.

Azienda Ospedaliera, Mantova, Itália: a maior instituição de saúde de Mantova, possuindo seis hospitais. OpenBSD serve para interligar o hospital principal de Mantova, funcionando como firewall.

Capitol College: única faculdade independente em Maryland dedicada à engenharia, ciência da computação, tecnologia da informação e negócios. Utiliza OpenBSD para uma variedade de funções, servindo seu *website*, protegendo a rede como *firewall*, sistema de detecção de intruso e hospedagem da sua autoridade certificadora interna.

Helsinki University Central Hospital, na Finlândia: Hospital Distrital que oferece atendimento médico especializado. OpenBSD é usado com DNS, *gateway* de e-mail, VPN e *firewall*.

Adobe Systems: Gigante do software. Utiliza OpenBSD como firewall em suas redes e sistemas de teste.

Network Security Technologies, Inc.: Empresa de segurança da informação localizada em Washington DC. Usa OpenBSD para detecção de intrusão de alta velocidade, VPN e aplicações de data warehousing. Usa OpenBSD em locais militares não revelados e vários órgãos do governo.

A lista dos locais que afirmam utilizar OpenBSD em seu ambiente é muito grande e pode ser conferida no site oficial do OpenBSD (OPENBSD, 2013).

4.2 SNORT

Snort é um sistema de detecção e prevenção a intrusão de rede. É desenvolvido pela *Sourcefire* e seu código é aberto. Combinando os benefícios de assinatura, protocolo e inspeção baseada em anomalia, Snort é a tecnologia de IDS/IPS mais implantada em todo o mundo, com milhões de downloads e aproximadamente 400.000 usuários registrados (SNORT, 2013).

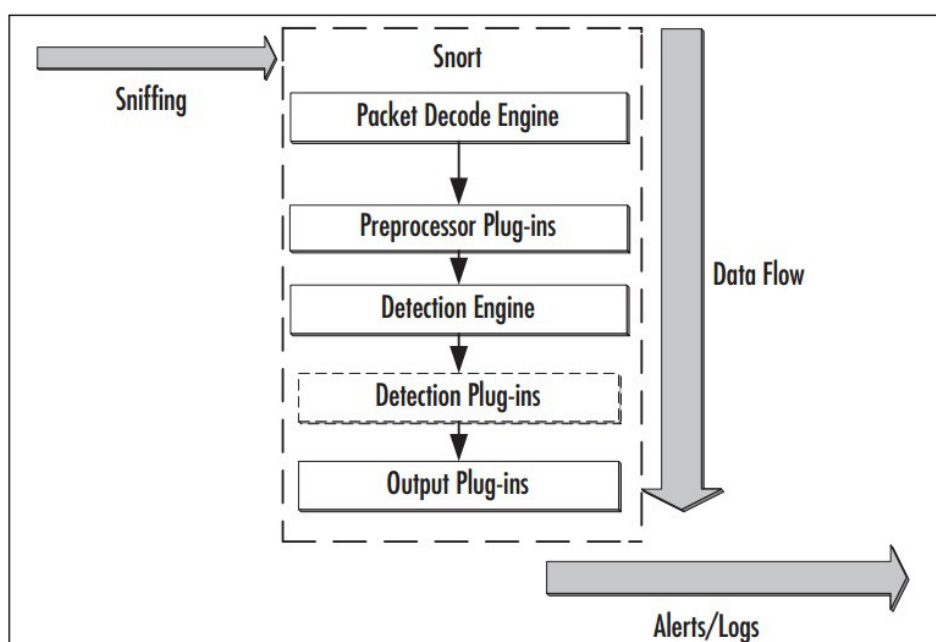
Seu desenvolvimento e atualização são constantes, tanto as regras de detecção quanto o código como um todo, são atualizados diariamente. Os módulos que o compõe são poderosas ferramentas, com capacidade de produzir uma quantidade enorme de informações sobre os ataques monitorados. Permite o monitoramento de tráfego de pacotes em rede IP, realizando análises em tempo real sobre diversos protocolos (SNORT BRASIL, 2013).

Possui basicamente três funções principais, que são: capacidade de servir como um *sniffer* de pacote, um registrador de pacotes ou um Sistema de Detecção de Intrusão baseado em rede (NIDS).

4.2.1 Recursos do SNORT

A Figura 6, ilustra os componentes presentes no SNORT:

Figura 6: Componentes do SNORT.



1. *Packet capture/decoder engine* (Mecanismo de captura / decodificador de pacotes): Primeiramente, o tráfego de rede é capturado através da biblioteca ***libpcap***. Os pacotes passam através do ***decode engine***, preenchendo a estrutura dos pacotes para os protocolos de enlace, onde serão decodificados para os protocolos de nível mais alto como as portas TCP e UDP (BEALE, 2003).

A decodificação é organizada ao redor das camadas da pilha do protocolo TCP/IP, onde as rotinas de decodificação são chamadas em ordem, partindo do nível de dados e subindo para o nível de transporte até terminar no nível de aplicação (SNORT BRASIL, 2013).

Ao entrar pela interface de rede, os pacotes são decodificados, determinando qual protocolo está sendo utilizado pelo pacote e se os dados coincidem com o comportamento que não é permitido para os pacotes desse protocolo. O decodificador de pacotes pode então gerar alertas tendo como base os cabeçalhos mal formados, pacotes excessivamente longos ou que possuam opções incomuns definidas no cabeçalho (BEALE, 2003, 2004).

2. *Preprocessor plug-ins* (Plugins de pré-processamento): Os pacotes são enviados através de um conjunto de pré-processadores, onde são analisados e manipulados antes de serem entregues a ***Detection Engine***. Cada pré-processador irá verificar se o pacote é algo que deva examinar, alertar sobre ou modificar (BEALE, 2003).

3. *Detection Engine* (Mecanismo de detecção): Os pacotes são então enviados para a ***Detection Engine***, onde irá verificar cada pacote com relação as diversas opções listadas nos arquivos de regras do Snort, através de testes individuais. Cada uma das opções de palavra-chave na regra, está ligada a um *plugin* de detecção, permitindo realizar testes adicionais (BEALE, 2003).

Este mecanismo é provavelmente o que primeiro vem a mente quando se imagina o funcionamento de um Sistema de Detecção de Intrusão. Este componente do Snort é o responsável por levar os dados do decodificador de pacotes e compará-los com as regras contidas no arquivo de configuração do Snort. Inicialmente, o mecanismo de detecção tentará determinar o que a regra definida deverá fazer, quando comparada com um determinado pedaço de dados. Ele então classifica primeiro por

protocolo e logo em seguida, através da identificação de características dentro do protocolo, por exemplo quando TCP e UDP é o número da porta de origem e destino, para ICMP, é o tipo do ICMP. Uma vez que o conjunto de regras foi determinado, o mecanismo de detecção segue seus procedimentos baseando-se na única regra relevante (BEALE, 2003, 2004).

4. Output plug-ins (Plugins de saída): O Snort gera os alertas a partir do mecanismo de detecção, pré-processadores ou do mecanismo de decodificação (BEALE, 2003).

5. Alerts / Logs (Alertas e Registros): Este componente é selecionado em tempo real com comandos condicionais de interrupção. O formato decodificado do registro, permite uma rápida análise dos dados armazenados pelo sistema, podendo ser deixados parcialmente incompletos para melhorar a performance (SNORT BRASIL, 2013).

O mecanismo de registro irá arquivar os pacotes que desencadearam as regras do Snort, enquanto o mecanismo de alerta será usado para notificar o analista sobre o acionamento de determinada regra (BEALE, 2003, 2004).

5. AMBIENTE DE TESTES

5.1. Componentes

Todo o ambiente foi criado utilizando Maquinas Virtuais, pois permite simular uma grande variedade de ambientes, sem a necessidade de adquirir equipamentos físicos.

O *VmWare Player*, versão 5.0.2, foi utilizado para criação das maquinas virtuais. Está é uma versão gratuita e pode ser baixada diretamente do site oficial¹. O ambiente virtual consistirá de um *Firewall*, um Servidor *Web* que irá sofrer o ataque de negação de serviço, uma estação de trabalho na rede interna que será responsável pelo monitoramento e uma estação de trabalho que irá simular o ataque ao servidor *web*.

Para a simulação ficar o mais próximo da realidade, foram criados segmentos de rede, separando a rede interna da rede de servidores.

5.1.1 Firewall

Nesse servidor será instalado o OpenBSD, que servirá de *firewall* para a rede. O sistema de detecção de intrusão, SNORT, também será instalado nesse servidor. Como foi visto no capítulo referente aos sistemas de detecção de intrusão, no que diz respeito ao posicionamento dos mesmos, a instalação do IDS nesse ponto é uma boa alternativa, pois permite a captura de uma grande quantidade de dados. A instalação do IDS em um outro servidor, separadamente do *firewall*, também poderia ter sido realizada, desde que o IDS ficasse no mesmo seguimento de rede do *firewall*. Tal procedimento não foi realizado devido ao aumento considerável da complexidade da criação do ambiente.

Para tornar a administração do IDS mais fácil, será instalado o BASE, que nada mais é do que uma interface web bastante intuitiva, que irá auxiliar o administrador, trazendo as informações geradas pelo SNORT em um formato mais amigável. Toda a instalação e configuração do BASE, juntamente com as suas dependências estão detalhadas no ANEXO I.

¹<http://www.vmware.com>

Configuração do Firewall

- Sistema Operacional: OpenBSD 5.3 (32-bits)
- Memória RAM: 512MB
- Processadores Virtuais: 2vCPUs
- Disco Rígido: 20GB
- 3 Interfaces de Rede:
 - Rede WAN (Internet - FALSA); IP 192.168.X.X/24
 - Rede LAN (Estação de administração); IP 10.10.10.1/24
 - Rede DMZ (Servidores); IP 172.16.10.1/24

O SNORT será instalado utilizando as ferramentas de instalação do próprio OpenBSD, já que o pacote de instalação está disponível em seu repositório oficial. Nada impede que o SNORT seja instalado a partir de seu código fonte, porém esse procedimento seria irrelevante para que o objetivo fosse alcançado. A instalação e configuração do SNORT, juntamente com suas dependências, estão listadas no ANEXO I.

5.1.2 DMZ (Zona Desmilitarizada) e Servidor Web

Para tornar a simulação o mais parecida possível com um ambiente real, foi criada uma DMZ, que está presente em praticamente todos os ambientes de rede. A rede DMZ é uma forma de separar os serviços que estarão disponíveis através da internet, sem comprometer a segurança da rede interna.

Para a DMZ, será utilizado o endereçamento 172.16.10.0/24. Nessa sub-rede será implantado um servidor *WEB*, cujo endereço IP será 172.16.10.10/24. Este servidor será o que irá sofrer um ataque de negação de serviço.

Configurações do servidor WEB:

- Sistema Operacional: Linux CentOS 6.4 (32bits).
- Servidor Web Apache 2
- Memória RAM: 1GB
- Processadores Virtuais: 2vCPUs
- Disco Rígido: 10GB
- Uma interface de rede

5.1.3 Rede Interna e Estação de Administração

A rede interna será identificada pelo endereçamento 10.10.10.0/24. Separada da DMZ, uma estação de administração ficará responsável pela gerência do ambiente como um todo. A administração do *Firewal*, SNORT, Alertas e gerenciamento do servidor *Web*, serão realizados por meio desta estação. Seu endereço será 10.10.10.10/24. O sistema operacional desta estação é indiferente, porem será utilizada uma instalação simples de uma distribuição Linux.

Configurações da estação de administração.

- Sistema Operacional: GNU/Linux Debian 6.0 – squeeze (32bits)
- Memória RAM: 512MB
- Processadores Virtuais: 1vCPU
- Disco Rígido: 10GB
- Uma interface de rede.

5.1.4 Estação do Atacante (Simulação do Ataque)

A estação que será utilizada para realizar o ataque, também será uma máquina virtual. Em um ataque de Negação de Serviço real, normalmente seriam utilizados várias estações espalhadas, conhecidas com *zumbis*. Para essa simulação, veremos que uma única estação será capaz de tornar o serviço indisponível.

Configuração da estação do atacante.

- Sistema Operacional: GNU/Linux Debian 6 – squeeze (32bits)
- Memória RAM: 512MB
- Processadores Virtuais: 1vCPU
- Disco Rígido: 10GB
- Uma interface de rede.

5.1.5 Ferramenta de Ataque DoS

A ferramenta escolhida para simular o ataque foi o **Slowloris HTTP DoS**. Aparentemente é uma ferramenta bastante simples, escrita em linguagem PERL, mas que consegue causar enormes dores de cabeça. Durante o ataque, quando bem sucedido, a página do servidor *web* fica indisponível enquanto que a utilização

de memória e CPU do servidor permanecem normais, o que dificulta a detecção do problema.

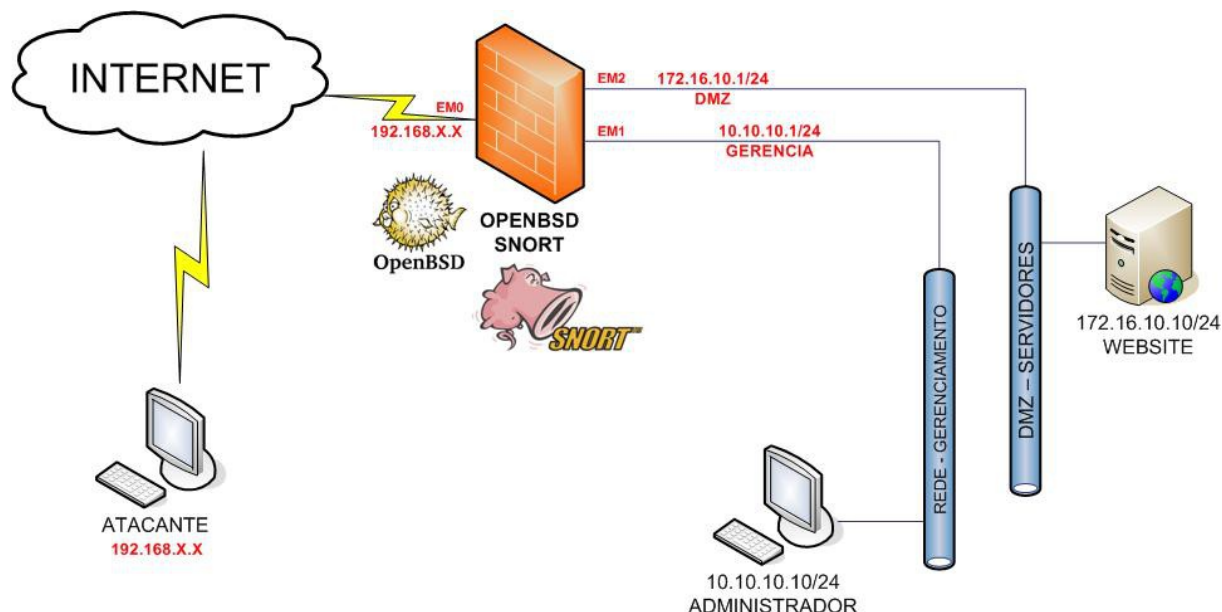
O *Slowloris*, consegue manter conexões abertas por meio do envio de solicitações HTTP parciais e permanece enviando cabeçalhos subsequentes em intervalos regulares. A ferramenta está disponível para download no site do desenvolvedor².

²<http://hackers.org/slowloris/>.

6 OPERACIONALIZAÇÃO DOS TESTES

6.1 Ambiente

Figura 7: Ambiente de simulação do ataque.



Fonte: próprio autor.

Na Figura 7, podemos visualizar a estrutura da rede de uma forma mais detalhada. Podemos ver, que o endereçamento de rede da estação do atacante e da interface WAN do *Firewall*, estão aparentemente no mesmo seguimento. Essa forma de endereçamento foi proposital, pois é necessário que haja comunicação da estação do atacante com o *Firewall*. Qualquer requisição que chegar até a interface WAN do *Firewall* será filtrada pelo mesmo e todos os pacotes serão analisados pelo SNORT, permitindo a detecção do ataque.

As interfaces WAN, LAN (gerencia) e DMZ do *Firewall*, podem ser identificadas como **em0**, **em1** e **em2** respectivamente. Podemos ver que, para alguma requisição chegar ao servidor *WEB*, será preciso primeiro passar pelo *Firewall* e consequentemente será analisada pelo SNORT.

O objetivo é que o SNORT consiga detectar o ataque, assim que a ferramenta for executada a partir da estação do atacante em direção ao servidor *Web*.

No ANEXO I estão descritos os detalhes de configuração do ambiente e as ferramentas adotadas.

6.2 Metodologia dos Testes

O Teste consistirá em realizar um ataque de negação de serviços, a partir da estação do atacante, utilizando a ferramenta de *DoS Slowloris*. Esta ferramenta, que possibilita realizar um ataque de negação de serviço, funciona enviando várias solicitações ao servidor, até que este não consiga mais atender a nenhuma das solicitações.

O ataque, vindo da estação do atacante, tendo como alvo o servidor *WEB*, localizado na DMZ da rede utilizada para simular o ataque, deverá deixar a página totalmente indisponível, durante todo o tempo em que o ataque permanecer ativo.

6.3 Resultados a Serem Alcançados

O objetivo principal não é conter o ataque, mas conseguir detectá-lo e consequentemente, alertar os administradores sobre o incidente, para que seja possível tomar alguma medida.

Logo abaixo, pode-se ver a regra que possibilita a detecção do ataque realizado pela ferramenta *Slowloris*.

```
alert tcp any any -> any any (msg:"Possivel Ataque de Negação de Serviço";
threshold: type threshold, track by_src, count 100 , seconds 5;
classtype:misc-activity; sid:2013; rev:1;)
```

A linha **alert tcp any any -> any any**, faz com que o SNORT analise o tráfego que entra ou sai de qualquer interface de rede, permitindo que o ataque seja detectado mesmo que este venha da rede interna.

A linha **threshold: type threshold, track by_src, count 100 , seconds 5**, significa 100 conexões a cada 5 segundos, ou seja, quando o SLOWLORIS começar a agir, ele vai justamente realizar várias conexões ao servidor em um intervalo de tempo bastante curto, o que consequentemente irá fazer com que o o SNORT gere um alerta imediatamente.

Baseando-se na regra acima, o SNORT deverá conseguir detectar o ataque, e as informações referentes ao mesmo deverão ser mostradas na interface do BASE. A interface do BASE facilita e auxilia na administração do IDS, trazendo infor-

mações em um formato mais legível para o administrador. O alerta sobre o possível ataque, deverá ser mostrado na interface do BASE em tempo real.

6.4 Execução dos Testes

Os testes foram iniciados a partir da estação do atacante, executando o comando que realiza o ataque, direcionado ao servidor *Web*.

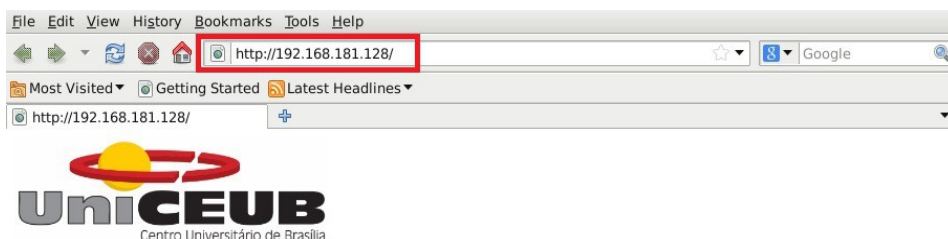
Comando para realizar o ataque:

```
perl slowloris.pl -dns 192.168.181.128 -port 80 -timeout 5 -num 600 -tcpto 5
```

O comando acima, realiza uma requisição ao servidor, na porta 80, com 600 *sockets* a cada 5 segundos.

A Figura 8 mostra o estado no qual a página se encontrava, antes de iniciar o ataque.

Figura 8: Acessando o Site a partir da estação do atacante



PÓS-GRADUAÇÃO LATO SENSU
Rede de Computadores com Ênfase em
Segurança

Ambiente de Testes

Simulação Denial Of Service

Fonte: próprio autor.

6.5 Resultado dos Testes

Antes do teste ser iniciado, o processamento e a utilização de memória do servidor web foram analisados, para que ambos pudessem ser utilizados como indicadores e com isso, definir o que é um comportamento normal ou anormal, durante ou depois a realização de um ataque de Negação de Serviços, permitindo dessa forma, identificar alguma alteração de comportamento no servidor. Na Figura 9, pode-se ver o consumo de recursos do servidor antes do ataque. É possível ver, que tanto o processamento quanto a utilização de memória estão baixos.

Figura 9: Servidor Web Antes do Ataque - Saída do comando HTOP

1 []1

2 []1

Mem[]1

Swp[]1

Tasks: 28, 4 thr: 1 running

Load average: 0.00 0.00

Uptime: 00:05:25

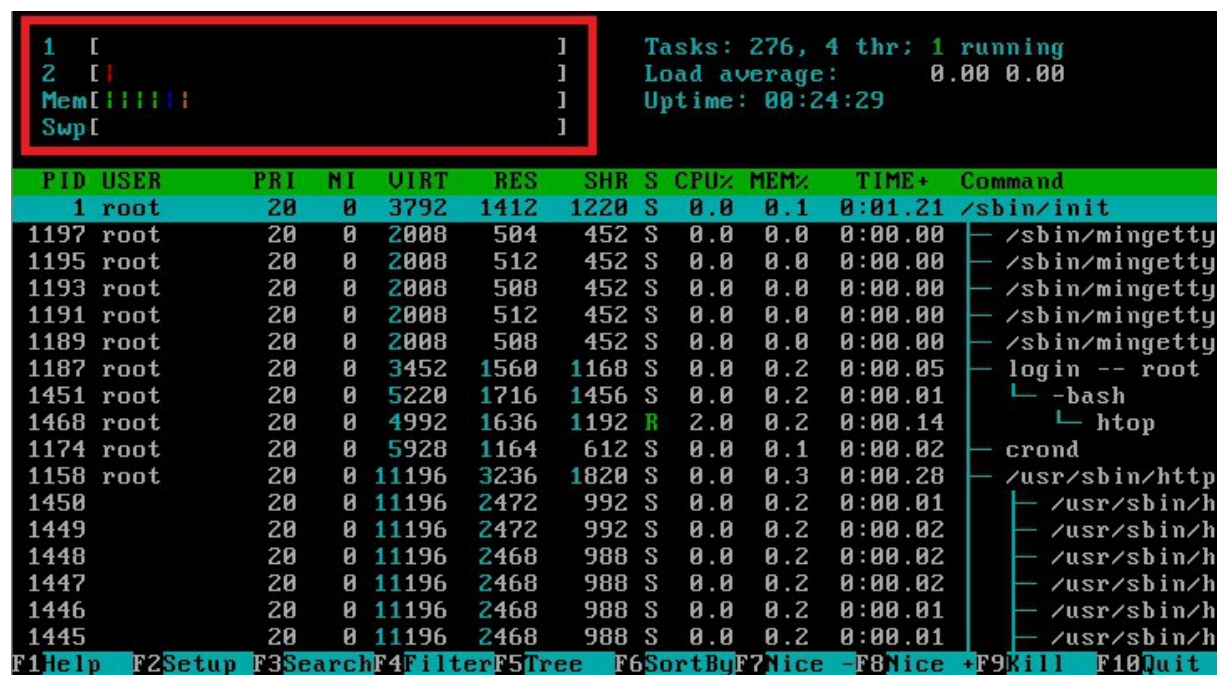
PID	USER	PRI	NI	UIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	2900	1412	1220	S	0.0	0.1	0:01.15	/sbin/init
1246	root	20	0	2008	512	452	S	0.0	0.0	0:00.00	/sbin/minigetty
1244	root	20	0	2008	508	452	S	0.0	0.0	0:00.00	/sbin/minigetty
1242	root	20	0	2008	508	452	S	0.0	0.0	0:00.00	/sbin/minigetty
1239	root	20	0	2008	508	452	S	0.0	0.0	0:00.00	/sbin/minigetty
1237	root	20	0	2008	504	452	S	0.0	0.0	0:00.00	/sbin/minigetty
1235	root	20	0	3452	1564	1168	S	0.0	0.2	0:00.05	login -- root
1247	root	20	0	5224	1712	1456	S	0.0	0.2	0:00.01	└ -bash
1262	root	20	0	4992	1544	1188	R	0.7	0.1	0:02.72	└ htop
1222	root	20	0	5932	1172	612	S	0.0	0.1	0:00.00	crond
1206	root	20	0	11196	3216	1804	S	0.0	0.3	0:00.02	/usr/sbin/httpd
1217		20	0	11196	2080	640	S	0.0	0.2	0:00.00	└ /usr/sbin/h
1216		20	0	11196	2080	640	S	0.0	0.2	0:00.00	└ /usr/sbin/h
1215		20	0	11196	2080	640	S	0.0	0.2	0:00.00	└ /usr/sbin/h
1213		20	0	11196	2080	640	S	0.0	0.2	0:00.00	└ /usr/sbin/h
1212		20	0	11196	2080	640	S	0.0	0.2	0:00.00	└ /usr/sbin/h
1211		20	0	11196	2080	640	S	0.0	0.2	0:00.00	└ /usr/sbin/h

F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit

Fonte: próprio autor.

Logo que iniciou-se a simulação do ataque, dentro de alguns segundos a página ficou indisponível. Durante esse tempo, os recursos do servidor ficaram sendo monitorados, na tentativa de detectar algum comportamento que pudesse caracterizar um ataque de Negação de Serviços. A figura abaixo demonstra os recursos do servidor durante o ataque.

Figura 10: Servidor Web Durante o Ataque - Saída do comando HTOP



Fonte: próprio autor.

Ao comparar a Figura 10 com a Figura 9, que mostra os recursos do servidor web antes de ser vítima de um ataque, pode-se constatar, que praticamente não houve alteração em seu processamento e utilização de memória, que como pode ser visto, permaneceu abaixo da metade da quantidade total disponível. Ao se observar e comparar o campo *Tasks* em ambas as imagens, pode-se ver que esta foi a única alteração de valores significativa. Antes do ataque, havia apenas 28 processos em execução e durante o ataque, a quantidade de processos aumentou consideravelmente, elevando-se para pouco mais de 270 processos em execução.

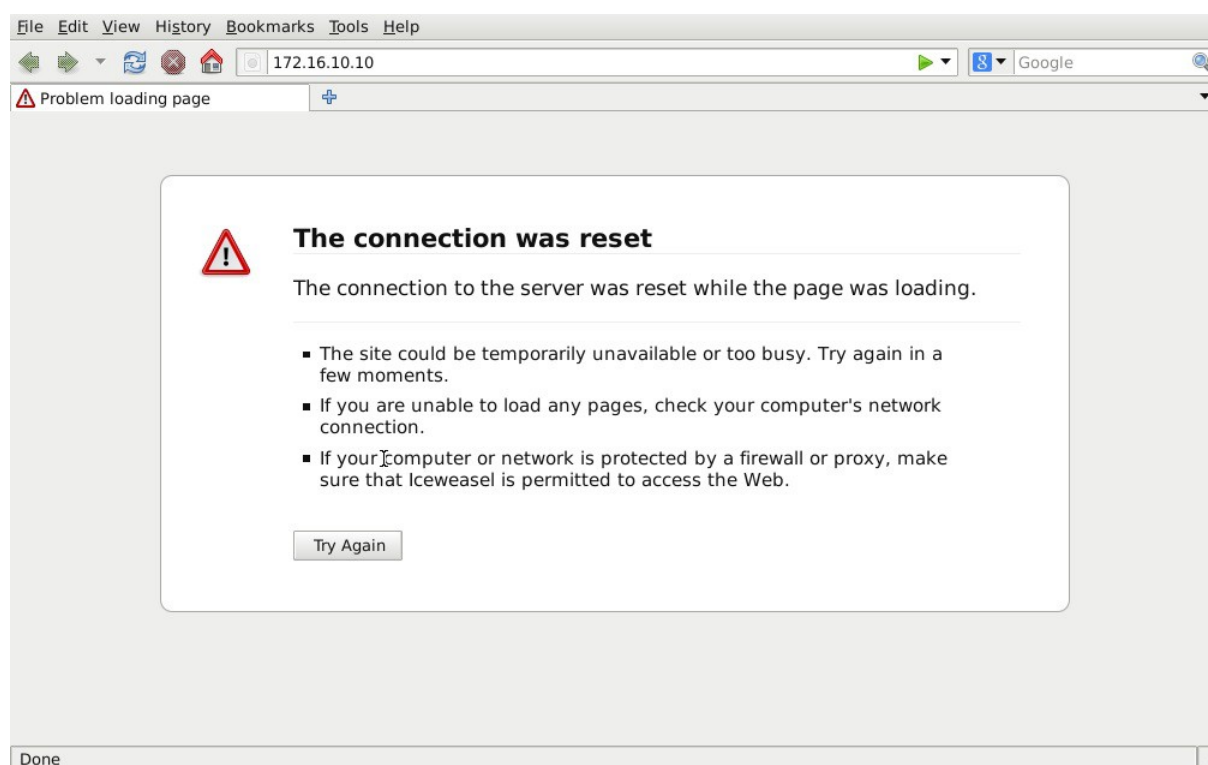
Porém, este comportamento, não necessariamente significaria que o servidor estaria sendo vítima de um ataque de Negação de Serviços. Poderia ser apenas um aumento na quantidade de visitantes ao site, já que nesse caso, não seria possível distinguir se eram acessos de usuários legítimos ou um ataque consumindo recursos do servidor.

Pode-se ver claramente, que a utilização de memória e CPU se manteve normal e que em momento algum durante o ataque, o servidor, a máquina virtual em si, apresentou travamento ou lentidão. Esse comportamento, dificulta ainda mais em um diagnóstico, pois o mais comum em um ataque de negação de serviços, é consumir recursos do servidor. Com o *Slowloris* isso não ocorre, devido ao fato da ferra-

menta explorar o serviço disponibilizado, que nesse caso, é o *Apache*, deixando-o impossibilitado de continuar respondendo às solicitações.

A Figura 11 mostra o acesso ao site totalmente indisponível. Pode-se observar na imagem, que a tentativa de acesso teve origem a partir da rede interna (Sub-rede 172.16.10.x só é acessível através da rede interna/gerencia), para que não houvesse dúvida que apenas a estação do atacante ficou impossibilitada de acessar o site.

Figura 11: Site totalmente inoperante durante o ataque.

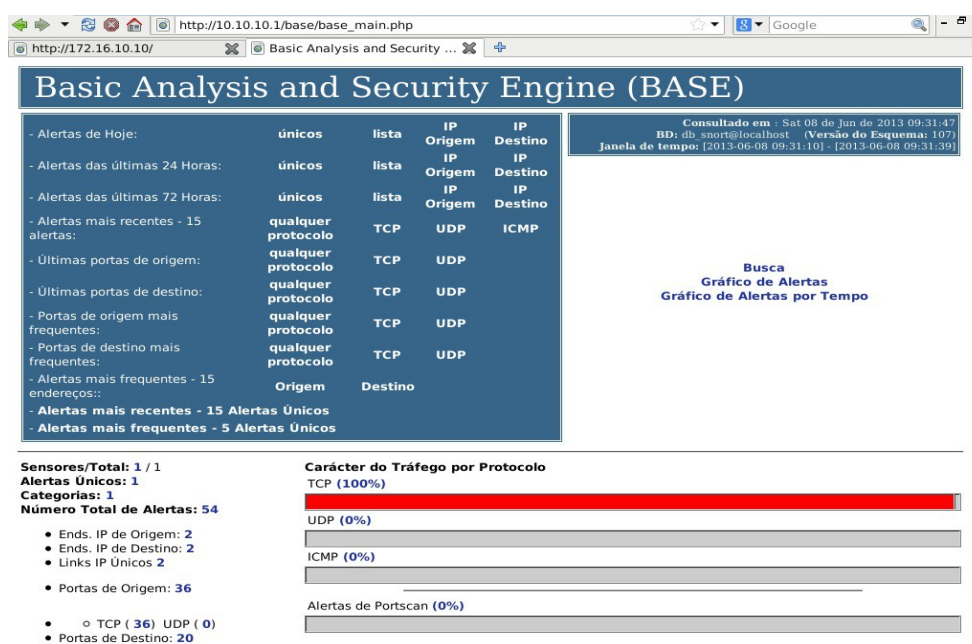


Fonte: próprio autor.

Mesmo que o serviço do Apache seja reiniciado várias vezes, o site ainda permanecerá lento ou completamente indisponível, enquanto durar o ataque. Quando o analista for verificar o servidor, verá que seu processamento e consumo de memória estão normais. Nesse cenário, dificilmente o analista pensaria, em um primeiro momento, que estaria sendo vítima de um ataque de negação de serviço, pois o comportamento mais comum nesses ataques, é justamente consumir recursos do servidor.

Em uma situação complicada como essa, podemos ver a importância que um Sistema de Detecção de Intrusão possui em uma rede. No ambiente, o SNORT, tendo como base a assinatura criada e explicada no item 6.3, conseguiu detectar o ataque e gerar um alerta para o Analista, que baseado nas informações trazidas pelo IDS, poderá tomar alguma atitude para resolver o problema. As imagens abaixo, mostram a detecção do ataque.

Figura 12: Tela principal do BASE - Ataque Detectado





Fonte: próprio autor.

Através da interface web do BASE, que trouxe as informações obtidas através do SNORT, podemos localizar a origem do ataque, mostrando justamente o endereço IP da estação do atacante.

Na Figura 13, pode-se ver algumas informações, referentes ao ataque detectado. Destaque para o campo **Assinatura**, que mostra a assinatura responsável por ter gerado o alerta. Ao observar a regra criada no item 6.3, a assinatura criada recebeu um **SID** (*Identificador* - **sid:2013;rev:1;**). Essa informação pode ser confirmada no campo assinatura, cujo resultado foi: **[snort] Snort Alert [1:2013:0]**. Ainda na Figura 13, pode-se ver o endereço de origem, que neste caso, foi apenas um, justamente o endereço da estação do atacante e também a porta para o qual o ataque foi direcionado (*porta 80 HTTP*).

Figura 13: Origem do Ataque

 http://10.10.10.1/base_main.php?num_result_rows=1&submit=Pesqui  Google

<http://172.16.10.10/> Basic Analysis and Security

Exibindo alertas 1-48 de 54 total

ID	< Assinatura >	< Data >	< End. de Origem >	< End. de Destino >	< Proto. Camada 4 >
#0-(1-46) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.128:80	192.168.181.131:39579	TCP
#1-(1-47) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.131:39586	192.168.181.128:80	TCP
#2-(1-48) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.131:39607	192.168.181.128:80	TCP
#3-(1-49) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.128:80	192.168.181.131:39640	TCP
#4-(1-50) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.131:39683	192.168.181.128:80	TCP
#5-(1-51) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.131:39783	192.168.181.128:80	TCP
#6-(1-52) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.131:39931	192.168.181.128:80	TCP
#7-(1-53) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.128:80	192.168.181.131:39988	TCP
#8-(1-54) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:39	192.168.181.131:39911	192.168.181.128:80	TCP
#9-(1-45) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:38	192.168.181.131:39919	192.168.181.128:80	TCP
#10-(1-25) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39588	192.168.181.128:80	TCP
#11-(1-26) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39606	TCP
#12-(1-27) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39619	192.168.181.128:80	TCP
#13-(1-28) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39655	192.168.181.128:80	TCP
#14-(1-29) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39654	TCP
#15-(1-30) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39684	192.168.181.128:80	TCP
#16-(1-31) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39708	TCP
#17-(1-32) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39717	192.168.181.128:80	TCP
#18-(1-33) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39760	192.168.181.128:80	TCP
#19-(1-34) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39759	TCP
#20-(1-35) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39794	192.168.181.128:80	TCP
#21-(1-36) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39801	192.168.181.128:80	TCP
#22-(1-37) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39557	TCP
#23-(1-38) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39585	TCP
#24-(1-39) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39753	192.168.181.128:80	TCP
#25-(1-40) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39858	192.168.181.128:80	TCP
#26-(1-41) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39859	TCP
#27-(1-42) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39893	192.168.181.128:80	TCP
#28-(1-43) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.128:80	192.168.181.131:39909	TCP
#29-(1-44) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:35	192.168.181.131:39926	192.168.181.128:80	TCP
#30-(1-18) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:21	192.168.181.131:39228	192.168.181.128:80	TCP
#31-(1-19) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:21	192.168.181.128:80	192.168.181.131:39237	TCP
#32-(1-20) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:21	192.168.181.131:39268	192.168.181.128:80	TCP
#33-(1-21) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:21	192.168.181.131:39280	192.168.181.128:80	TCP
#34-(1-22) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:21	192.168.181.131:39421	192.168.181.128:80	TCP
#35-(1-23) [snort]	Snort Alert [1.2013:0]	2013-06-08 09:31:21	192.168.181.131:39524	192.168.181.128:80	TCP

Fonte: próprio autor.

Ao clicar sobre o alerta, a tela abaixo será apresentada, trazendo informações mais detalhadas.

Figura 14: Origem do Ataque - Informações Detalhadas

[http://10.10.10.1/base/base_qlert.php?submit=%230-\(1-54\)&sort_order=](http://10.10.10.1/base/base_qlert.php?submit=%230-(1-54)&sort_order=)
<http://172.16.10.10/>
[Basic Analysis and Security ...](#)

ID #	Tempo	Assinatura que despoletou
1 - 54	2013-06-08 09:31:39	[snort] Snort Alert [1:2013:0]

Meta

Sensor	Sensor Endereço	Interface	Filtro
	firewall.localdomain:em0	em0	nenhum

Grupo de Alertas nenhum

End. de Origem	End. de Destino	Ver	Hdr Len	TOS	comprimento	ID	fragment	offset	TTL	chksum
192.168.181.131	192.168.181.128	4	20	0	64	31151	no	0	64	54451 = 0xd4b3

Opções nenhum

Orig Porta	Dest Porta	R 1	R 0	U R G	A C K	P S S	R S Y	F I N	seq #	ack	offset	res	window	urp	chksum
39911 [sans] [tanto] [sstats]	80 [sans] [tanto] [sstats]				X				4192069053	1798721721	44	0	183	0	24455 = 0x5f87

TCP

	código	comprimento	dados
Opções	#1 (1) NOP	0	
	#2 (1) NOP	0	
	#3 (8) TS	8	000FA0CF003EA354
	#4 (1) NOP	0	
	#5 (1) NOP	0	
	#6 (5) SACK	8	6B3650B86B3650B9

Fonte: próprio autor.

Na Figura acima, é possível ver mais algumas informações referentes ao ataque. Além do endereço de origem, é possível identificar o **Sensor** que detectou o ataque, que neste caso era o próprio *Firewall* e também a interface de rede. Informações mais a nível de pacote também podem ser facilmente identificadas, como por exemplo o tamanho do cabeçalho e o tempo de vida do pacote.

CONCLUSÃO

Manter uma rede segura não é uma tarefa fácil, todos os dias surgem novas ameaças e vulnerabilidades, que caso não sejam tratadas corretamente poderão causar enormes prejuízos.

Os ataques de negação de serviço tem sido um grande problema para as organizações, principalmente quando a organização possui toda a sua área de negócio voltada para a Internet.

Este estudo acadêmico permitiu compreender as diferentes formas de realizar um ataque de negação de serviço. Viu-se que não existe uma forma única de realizar esse tipo de ataque e que é possível não apenas explorar alguma vulnerabilidade em determinado servidor, mas também explorar vulnerabilidades nos protocolos de comunicação, o que dificulta ainda mais encontrar uma solução definitiva para o problema.

Foi possível implantar um ambiente, que permitiu compreender de forma prática, como é o funcionamento de um ataque de negação de serviços. Viu-se que uma ferramenta relativamente simples, foi capaz de deixar um site totalmente inoperante. Durante o ataque, foi possível mostrar, que, se não houvesse um Sistema de Detecção de Intrusão, o analista poderia perder bastante tempo tentando encontrar a causa do problema.

Pôde-se ver a importância que um sistema de detecção de intrusão possui dentro de uma organização. Foi possível constatar que, mesmo a rede sendo vítima de um ataque, o IDS conseguiu gerar um alerta, facilitando a tomada de uma decisão e consequentemente, reduzir o tempo que o serviço poderia ficar indisponível.

Muitas organizações tem receio de utilizar software livre, ainda mais em pontos críticos de sua rede. Porém, pode-se constatar a eficiência do SNORT e espera-se que esse estudo seja útil para incentivar a utilização de sistemas de detecção de intrusão, baseados em software livre, visando principalmente amenizar os impactos causados por um ataque de negação de serviços.

Durante a realização deste trabalho, a partir dos testes e análises realizadas, algumas propostas para futuros trabalhos foram surgindo. Uma delas é a questão de não apenas detectar o ataque, mas conseguir por exemplo, bloqueá-lo automaticamente. Sistemas de Detecção de Intrusão, assim como os Ataques de Negação de Serviços, são assuntos fascinantes e ainda há bastante a se estudar sobre eles. Por isso, espera-se também, que este estudo acadêmico seja apenas um ponto de partida para novos trabalhos e futuras pesquisas dentro da área.

REFERÊNCIAS

ABNT. *NBR ISO/IEC 27002:2005*: Tecnologia da informação: Técnicas de segurança: Código de prática para a gestão da segurança da informação. Rio de Janeiro, 2005.

BEALE, Jay. *et al.* **Snort 2.0 Intrusion Detection**. Estados Unidos: Syngress Publishing, 2003.

BEALE, Jay. *et al.* **Snort 2.1 Intrusion Detection**. 2 ed. Estados Unidos: Syngress Publishing, 2004.

BRANDÃO, José Eduardo Malta De Sá. **Composições de IDSs**: Viabilizando o Monitoramento de Segurança em Ambientes de Larga Escala. 141f. Tese (Doutorado em Engenharia Elétrica) - Florianópolis: Universidade Federal de Santa Catarina, 2007. Disponível em: <<http://gcseg.das.ufsc.br/wssec/pubs/brandao07-composicoes-ids-tese.pdf>>. Acesso em Jul. 2013.

CAMPOS, André L. N. **Sistemas de Segurança da Informação**: Controlando os Riscos. Florianópolis: Visual Books, 2006. 180 p.

CERT. Coordination Center. **Denial of Service Attacks**. 2 Out. 1997. Disponível em: <http://www.cert.org/tech_tips/denial_of_service.html>. Acesso em: Mar. 2013.

FERREIRA, Fernando Nicolau Freitas. **Segurança da Informação**. Rio de Janeiro: Ciência Moderna, 2003.

HUSSAIN, Alefiya; HEIDEMANN, John; PAPADOPOULOS, Christos. **A Framework for Classifying Denial of Service Attacks** – Extended. 25 Jun. 2003. Disponível em: <<ftp://ftp.isi.edu/isi-pubs/tr-569b.pdf>>. Acesso em Dez. 2012.

LAUFER, Rafael P. *et al.* **Negação de Serviço**: Ataques e Contramedidas em Minicursos do Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Florianópolis, Brazil, pp. 1-63, Set. 2005. Disponível em: <<http://www.gta.ufjf.br/ftp/gta/TechReports/LMVB05a.pdf>>. Acesso em: Jan. 2013.

LEOBONS, Rodrigo Maestrelli. **IDS - Sistemas de Detecção de Intrusos**. Universidade Federal do Rio de Janeiro - Engenharia de Computação e Informação, 2007. Disponível em: <http://www.gta.ufjf.br/grad/07_2/rodrigo_leobons/deteccao.html>. Acesso em: Abr. 2013.

NAKAMURA, Emilio Tissato; GEUS, Paulo Lício de. **Segurança de redes em ambientes cooperativos**. São Paulo: Novatec, 2007.

OPENBSD. The OpenBSD project produces a FREE, multi-platform 4.4BSD-based UNIX-like operating system. Disponível em: <<http://www.openbsd.org/>>. Acesso em Jan. 2013.

SÊMOLA, Marcos. **Gestão da Segurança da Informação**: Uma visão Executiva. Rio de Janeiro: Elsevier, 2003 – 12ª reimpressão.

SNORT. Open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. Disponível em <<http://www.snort.org/>>. Acesso em Fev. 2013.

SNORT BRASIL. The Open Source Network Intrusion Detection System. Disponível em <<http://www.snort.com.br>>. Acesso em Mar. 2013.

SOUZA, Denis Augusto A. de. **FreeBSD**: O poder dos servidores em suas mãos. São Paulo. Novatec Editora, 2009.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de Computadores** 5. ed. São Paulo: Pearson Prentice Hall, 2011.

TRUSTWAVE'S SPIDERLABS. SpiderLabs is an elite team of ethical hackers, investigators and researchers at Trustwave advancing the security capabilities of leading businesses and organizations throughout the world. Disponível em <<http://blog.spiderlabs.com/>>. Acesso em Abr. 2013.

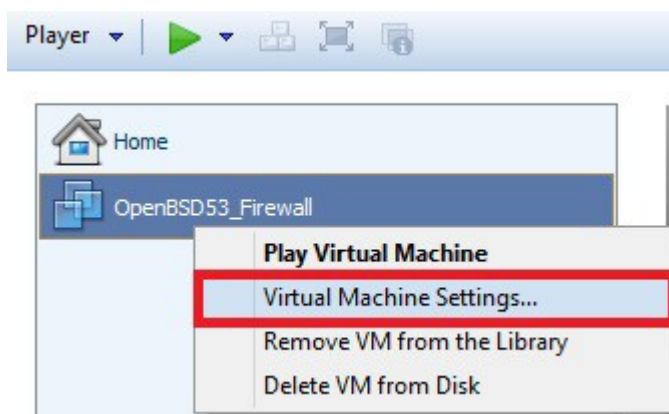
ANEXO I

A. IMPLANTAÇÃO E CONFIGURAÇÃO DO AMBIENTE VIRTUALIZADO

A.1 Configuração do Firewall

Abaixo podemos ver, como deve ser realizada a configuração das interfaces de rede do *Firewall*. Essa configuração pode ser realizada durante a criação da máquina virtual, ou mesmo depois de já ter instalado o sistema operacional. Aqui vamos presumir que o sistema operacional já está instalado e a máquina virtual está **desligada**.

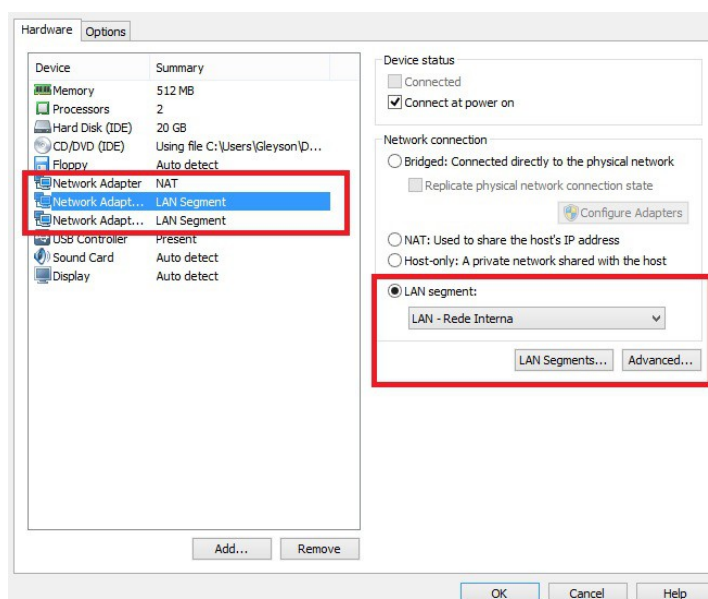
Figura 15: Configurações do Firewall



Fonte: próprio autor.

1) - Clicando com o botão direito do mouse sobre a máquina virtual, a janela mostrada na Figura 15, deverá aparecer. Clique em “**Virtual Machine Settings...**” para darmos início a configuração.

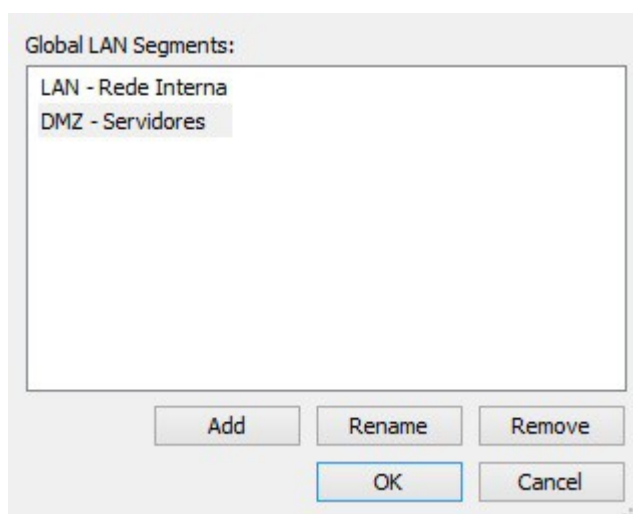
Figura 16: Configuração dos Segmentos de Rede - Firewall



Fonte: próprio autor.

2) – Veja que foram adicionadas 3 (três) interfaces de rede. A primeira interface deverá ser configurada como **NAT**. Esta interface será a de **WAN**. As outras duas interfaces serão as interfaces de **LAN** e **DMZ**. Para que seja possível isolar os dois segmentos de rede, vamos utilizar o recurso de “LAN Segments” disponível no VmWare Player. A Figura 17 mostra os dois segmentos criados.

Figura 17: Criando os Segmentos de Rede



Fonte: próprio autor.

Depois que tiver criado os seguimentos, basta associá-los as devidas interfaces de rede. É importante ficar atento nesse ponto, para não gerar confusão. Na dúvida, verifique qual o endereço MAC de cada interface, isso irá facilitar a configuração no sistema operacional. Para visualizar qual o endereço MAC da interface, basta clicar no botão “Advanced...”, mostrado na Figura 16. Na Figura 18, pode-se observar o MAC Address da interface configurada como “LAN – Rede Interna”.

Figura 18: Visualizando o MAC Address

The screenshot shows a network configuration window with three sections: 'Incoming Transfer', 'Outgoing Transfer', and 'MAC Address'. Both 'Incoming Transfer' and 'Outgoing Transfer' sections have 'Bandwidth' set to 'Unlimited', 'Kbps' set to an empty field, and 'Packet Loss (%)' set to '0.0'. The 'MAC Address' section has a text field containing '00:0C:29:26:3B:F1' and a 'Generate' button. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Fonte: próprio autor.

3) – Após configurar as interfaces de rede, a maquina virtual já pode ser ligada. A configuração das interfaces de rede podem ser verificadas através do comando *ifconfig*. A saída desse comando pode ser visualizado na Figura 19. Destaque para o campo que mostra o *MAC Address* da interface.

Figura 19: Saída do comando ifconfig

```

lladdr 00:0c:29:26:3b:e7
priority: 0
groups: egress
media: Ethernet autoselect (1000baseT full-duplex,master)
status: active
inet6 fe80::20c:29ff:fe26:3be7%em0 prefixlen 64 scopeid 0x1
inet 192.168.181.128 netmask 0xfffff00 broadcast 192.168.181.255
em1: flags=8002<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:0c:29:26:3b:f1
    priority: 0
    media: Ethernet autoselect (1000baseT full-duplex,master)
    status: active
em2: flags=8002<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:0c:29:26:3b:fb
    priority: 0
    media: Ethernet autoselect (1000baseT full-duplex,master)
    status: active
enc0: flags=0<>
    priority: 0
    groups: enc
    status: active
pflog0: flags=141<UP,RUNNING,PROMISC> mtu 33196
    priority: 0
    groups: pflog
# _

```

Fonte: próprio autor.

4) – Com as interfaces corretamente configuradas e detectadas pelo sistema operacional, o próximo passo agora é dar início as configurações dentro do sistema.

IMPORTANTE: *Todos os comandos deverão ser executados com o usuário **ROOT**.* Sempre que houver o sinal “\” (barra para esquerda), significa uma quebra de linha e que a linha de baixo é continuação do comando.

A. 1.1 Configurando o Sistema Operacional

Antes de mais nada, vamos precisar configurar o “repositório” que será utilizado para baixar e instalar todos os pacotes necessários para a configuração. No site do OpenBSD existem uma enorme variedade de repositórios. Nesse projeto foi utilizado o repositório localizado no Brasil. Abaixo podemos ver como configurar esse repositório:

```
echo "export PKG_PATH=http://openbsd.locaweb.com.br/5.3/packages/i386/" >> \
/root/.profile
```

É preciso configurar o endereço IP de cada interface. No projeto, as interfaces de rede ficaram definidas das seguintes formas:

EM0 – WAN

EM1 – LAN

EM2 - DMZ

Para realizar a configuração das interfaces de rede no OpenBSD, precisamos criar os arquivos referentes a cada uma delas. Dentro desses arquivos, pode-se definir o endereçamento da interface.

A interface WAN, irá receber seu endereço automaticamente, por isso que, nas opções do *VmWare*, essa interface foi configurada como NAT. Vamos precisar configurar apenas as interfaces de **LAN** e **DMZ**, que são **em1** e **em2** respectivamente.

Configuração das interfaces

EM1 – LAN

1. Criar o arquivo de configuração da interface “em1”:

```
touch /etc/hostname.em1
```

2. Editar o arquivo criado e inserir as configurações da interface:

```
inet 10.10.10.1 255.255.255.0
```

EM2 – DMZ

1. Criar arquivo de configuração da interface “em2”:

```
touch /etc/hostname.em2
```

2. Editar o arquivo criado e inserir as configurações da interface:

```
inet 172.16.10.1 255.255.255.0
```

Após concluir, use o comando abaixo para aplicar as alterações nas interfaces de rede:

```
sh /etc/netstart
```

Após executar o comando que aplica as configurações, as interfaces deverão estar corretamente configuradas. A configuração pode ser verificada através do comando **ifconfig**, conforme pode ser visto na Figura 20.

Figura 20: Saída do comando **ifconfig** - Configuração das interfaces

```

status: active
inet6 fe80::20c:29ff:fe26:3be7%em0 prefixlen 64 scopeid 0x1
inet 192.168.181.128 netmask 0xffffffff broadcast 192.168.181.255
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:0c:29:26:3b:f1
    priority: 0
    media: Ethernet autoselect (1000baseT full-duplex, master)
    status: active
    inet6 fe80::20c:29ff:fe26:3bf1%em1 prefixlen 64 scopeid 0x2
    inet 10.10.10.1 netmask 0xffffffff broadcast 10.10.10.255
em2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:0c:29:26:3b:fb
    priority: 0
    media: Ethernet autoselect (1000baseT full-duplex, master)
    status: active
    inet6 fe80::20c:29ff:fe26:3bf1%em2 prefixlen 64 scopeid 0x3
    inet 172.16.10.1 netmask 0xffffffff broadcast 172.16.10.255
enc0: flags=0<>
    priority: 0
    groups: enc
    status: active
pflog0: flags=141<UP,RUNNING,PROMISC> mtu 33196
    priority: 0
    groups: pflog
# _

```

Fonte: próprio autor.

O OpenBSD não vem configurado para trabalhar como gateway/roteador, então será necessário habilitar essa função.

Para habilitar a função de roteamento, será necessário editar o arquivo abaixo.

```
/etc/sysctl.conf
```

Descomente a seguinte linha (normalmente é uma das primeiras):

```
net.inet.ip.forwarding=1
```

Reinicie o servidor:

```
reboot
```

Verifique se a configuração está correta:

```
sysctl -a | grep net.inet.ip.forwarding
```

A saída do comando deverá ser:

```
net.inet.ip.forwarding=1
```

A. 1.2 Configurando as regras de filtragem

O OpenBSD utiliza o *Packet Filter* (PF), para realizar a filtragem de tráfego TCP/IP e também realizar a Tradução de Endereços de Rede (NAT). Possui a capacidade de realizar a normalização e condicionamento do tráfego TCP/IP, providenciando o controle de largura de banda e priorização de pacotes (OPENBSD)

Para configurar as regras de filtragem, será preciso editar o arquivo de configuração **/etc/pf.conf**. Edite-o, exclua todo o conteúdo do arquivo e insira as regras abaixo:

```
wan_if=em0
lan_if=em1
dmz_if=em2

set skip on lo

match out on egress inet from !(egress:network) to any nat-to (egress:0)

block in log
pass out quick
```

```
antispoof quick for { lo $lan_if $dmz_if }

block in on ! lo0 proto tcp to port 6000:6010

pass in on $lan_if from 10.10.10.10 to any
pass in on $wan_if proto tcp from any to any port 80 rdr-to 172.16.10.10
```

Essas regras irão bloquear todo o tráfego de entrada e realizará um redirecionamento para o servidor web. Caso esteja transferindo arquivos para dentro do servidor, é aconselhável que essas regras sejam aplicadas por último.

Para aplicar as regras, o comando abaixo deverá ser executado:

```
pfctl -f /etc/pf.conf
```

As regras podem ser habilitadas e desabilitadas a qualquer momento, utilizando os comandos abaixo:

Para Habilitar:

```
pfctl -e
```

Para Desabilitar:

```
pfctl -d
```

A. 2 Instalação e configuração do SNORT

Antes de iniciar a instalação do SNORT, será preciso instalar primeiro algumas dependências:

- banco de dados MySQL;
- servidor web APACHE-2;
- linguagem PHP-5
- PCRE – Biblioteca de Expressões Regulares
- Libdnet – API genérica de rede que permite acesso a vários protocolos
- Barnyard2 – Sistema que possibilita interpretar os dados do SNORT. Consegue ler o arquivo saída do SNORT em formato binário e reenviar os dados para um banco de dados.
- DAQ – API para aquisição de dados.

- BASE – Interface WEB para realizar a análise de intrusões detectadas pelo SNORT. Caso haja um ataque ou apenas a tentativa, o administrador poderá ser alertado através desta interface web.

A. 2.1 Instalação e configuração do MySQL-Server

Instalação

```
pkg_add mysql-server
```

Inicializar o diretório de dados do mysql e cria as tabelas do sistema

```
mysql_install_db
```

Se o passo anterior não for executado, não será possível iniciar o serviço do MySQL
Inciar o serviço:

```
/etc/rc.d/mysqld start
```

Utilize o comando a seguir para continuar com a configuração. Todos os passos são bastante intuitivos.

```
mysql_secure_installation
```

Nesse passo, será solicitado uma senha. Como o MySQL ainda não foi utilizado, ele ainda estará configurado com uma “*senha em branco*”.

Tecla “ENTER” e prossiga para configurar uma senha, em seguida, basta pressionar "ENTER" novamente para as outras perguntas.

Será preciso criar um usuário e uma base de dados, para ser utilizado pelo BASE e o BARNYARD2. Caso o BARNYARD2 não esteja corretamente configurado, o BASE não conseguirá mostrar os alertas do SNORT, mesmo que os ataques sejam detectados.

Criar usuário que será utilizado pelo base e barnyard2

Logue-se no banco. Será solicitada uma senha. Entre com a senha configurada anteriormente.

```
mysql -u root -p
```

Criar usuário utilizado pelo BASE. Nesse projeto, foi utilizado o usuário “**usr_snort**”. Nada impede que seja utilizado um outro nome de usuário.

Comando para criar usuário:

```
CREATE USER 'usr_snort'@'%' IDENTIFIED BY 'mono123';  
GRANT USAGE ON *.* TO 'usr_snort'@'%' IDENTIFIED BY 'mono123' WITH  
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UP-  
DATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

Comando para criar base de dados e conceder o acesso para o usuário criado anteriormente:

```
CREATE DATABASE db_snort;  
GRANT ALL PRIVILEGES ON `db\_snort` . * TO 'usr_snort'@'%' WITH GRANT  
OPTION;
```

A 2.2 *Instalação e configuração do apache*

```
pkg_add -i apache_httpd
```

O apache deverá estar configurado para “escutar” e responder apenas na interface de LAN. Para isso, vamos editar o arquivo de configuração do apache e realizar as devidas alterações.

Edite o arquivo de configuração:

```
/etc/apache2/httpd2.conf
```

Localize e altere/descomente as seguintes linhas:

```
ServerName 10.10.10.1  
Listen 10.10.10.1:80
```

A 2.3 Instalação e configuração do barnyard2

O Barnyard2 não está disponível nos repositórios do OpenBSD, então será preciso baixar diretamente do site do desenvolvedor e compilar o código.

Site Oficial: <http://www.securixlive.com/>

Link direto para download: <http://www.securixlive.com/barnyard2/download.php>

Descompactar o arquivo:

```
tar -zxvf barnyard2-1.9.tar.gz
```

Acessar o diretório descompactado

```
cd barnyard2-1.9
```

Iniciar configuração e compilação. Observe a opção “--with-mysql”. Caso essa opção não seja informada, o barnyard2 não vai conseguir acessar o banco de dados.

```
./configure --with-mysql
```

```
make
```

```
make install
```

Dentro do diretório do “barnyard2”, existe um diretório chamado “schemas/”. Esse diretório contém os *scripts* para criação dos bancos de dados.

Mude para o diretório “**schemas/**”

```
cd /schemas/
```

Execute o comando abaixo, para criar a estrutura de tabelas dentro do banco de dados, criado anteriormente para ser usado pelo BASE.

```
mysql -u root -p --database=db_snort -vv < create_mysql
```

Com o Barnyard2 instalado, precisamos agora editar o arquivo de configuração e realizar algumas alterações importantes para seu correto funcionamento.

Editar o arquivo de configuração do “barnyard2”

```
/usr/local/etc/barnyard2.conf
```

Realizar as seguintes alterações dentro do arquivo de configuração:

```
config interface: em0
config logdir: /var/snort/log/
output database: log, mysql, user=usr_snort password=mono123 dbname=db_snort
host=localhost
```

Ainda dentro do arquivo de configuração, comente (utilizando o sinal #) a seguinte linha:

```
#config sid_file: /etc/snort/sid-msg.map
```

Como foi dito anteriormente, o Barnyard2 não está disponível nos repositórios oficiais do OpenBSD, por isso foi preciso baixar o código fonte e compilar. Após o barnyard2 ter sido instalado e configurado, será preciso sempre rodar o comando manualmente para que ele funcione, o que não é algo interessante de ser fazer, ainda mais em um servidor. Por isso, vamos criar o script de inicialização para o serviço do barnyard2, para não termos que ficar iniciando o serviço manualmente toda vez.

Criar o arquivo de inicialização:

```
touch /etc/rc.d/barnyard2
```

Editar o arquivo criado:

```
/etc/rc.d/barnyard2
```

Inserir as linhas abaixo dentro do arquivo:

```
#!/bin/sh
#
daemon="/usr/local/bin/barnyard2 -D"
daemon_flags="-c /usr/local/etc/barnyard2.conf -d /var/log/snort/ -f snort.u2 -w /var/snort/snort_tmp"

. /etc/rc.d/rc.subr

rc_cmd $1
```

A 2.4 Instalação e configuração do BASE e suas dependências.

O BASE é uma interface Web bastante intuitiva. É por meio dela que conseguiremos visualizar de forma fácil, não apenas os ataques bem sucedidos, mas também as tentativas de ataque. O BASE permite a geração de gráficos, mostrando por exemplo informações referentes a hora do ataque e sua origem, entre outras informações que auxiliam o analista a tomar uma decisão.

Para que o BASE funcione de forma correta, será preciso instalar algumas dependências. Todas estas dependências serão instaladas no próprio *Firewall*, juntamente com o SNORT. Em um ambiente real, talvez a melhor opção fosse instalar o IDS em outro servidor, separadamente do *Firewall*, como foi dito no capítulo referente a IDS, mas para o objetivo pretendido, não haverá problemas em instalar o BASE no mesmo ambiente do *Firewall*.

Instalação das dependências:

PEAR

```
pkg_add pear-1.9.4p1
```

```
pear install Image_Canvas-0.3.5
```

```
pear install Image_Graph-0.8.0
```

PHP

```
pkg_add php-5.3.21-ap2
```

```
pkg_add php-mysql-5.3.21
```

Criar um link simbólico para a biblioteca do MySQL.

```
ln -sf /etc/php-5.3.sample/mysql.ini /etc/php-5.3/mysql.ini
```

```
pkg_add php-gd-5.3.21
```

Criar um link simbólico para a biblioteca do GD.

```
ln -sf /etc/php-5.3.sample/gd.ini /etc/php-5.3/gd.ini
```

Editar o arquivo de configuração do PHP:

```
/etc/php-5.3.ini
```

Localizar e alterar a opção "**error_reporting**" dentro do arquivo de configuração do PHP :

```
error_reporting = E_ALL & ~E_NOTICE
```

Após ter instalado o PHP, será preciso configurar o APACHE para funcionar corretamente.

Editar o arquivo de configuração do APACHE:

```
/etc/apache2/httpd.conf
```

Ao final do arquivo, inserir as linhas abaixo:

```
LoadModule php5_module /usr/local/lib/php-5.3/libphp5.so

<IfModule mod_php5.c>
    AddType application/x-httpd-php .php .phtml .php3
    AddType application/x-httpd-php-source .phps
# Most php configs require this
    DirectoryIndex index.php
</IfModule>
```

Baixar o ADODB - Dependência requerida pelo BASE

Site para Download: <http://adodb.sourceforge.net/>

Descompacte o arquivo baixado:

```
tar zxvf adodb518a.tgz
```

Em seguida, mova o diretório descompactado para o diretório de configuração do APACHE

```
mv adodb5 /etc/apache2/
```

Esse caminho será necessário durante a configuração do BASE.

Instalação do base

O download pode ser realizado através do site oficial: <http://base.secureideas.net/>

Apos baixar o BASE, descompacte-o e mova os arquivos para o diretório de publicação do apache:

```
tar zxvf base-1.4.5.tar.gz
```

```
mv base-1.4.5 /var/apache2/htdocs/
```

Para facilitar o acesso, renomeie o diretório do BASE.

```
cd /var/apache2/htdocs/
```

```
mv base-1.4.5 base
```

Durante a instalação, o BASE precisará alterar um arquivo de configuração, era para ser apenas um arquivo, porém mesmo alterando as permissões desse arquivo de configuração, o BASE continua alertando que não é possível gravar alterações no arquivo especificado. A solução para poder prosseguir, é alterar as permissões de todo o diretório e quando tiver finalizado a instalação, basta voltar com as permissões corretas.

Alterando as permissões do diretório

```
chmod -R 777 base/
```

A partir da estação de administração, acesse a interface do BASE para dar inicio a instalação.

```
http://10.10.10.1/base
```

Será mostrada a tela Inicial. Realiza uma checagem geral antes de prosseguir com a instalação. Se tudo estiver certo, a tela deverá parecer com a da Figura 21.

Figura 21: Início da Configuração do BASE

Basic Analysis and Security Engine (BASE) Setup Program

The following pages will prompt you for set up information to finish the install of BASE.
If any of the options below are red, there will be a description of what you need to do below the chart.

Settings	
Config Writeable:	Yes
PHP Version:	5.3.21
PHP Logging Level:	[ERROR][WARNING][PARSE]

Continue

Fonte: próprio autor.

A tela mostrada na Figura 22, solicita o caminho onde se encontra o ADODB. Se tiver seguido todos os passos corretamente, o caminho será: **/etc/apache2/adodb5**. Caso não seja especificado o caminho corretamente, o instalador apresentará um erro e não irá prosseguir com a instalação.

Figura 22: Configuração do caminho ADODB

Basic Analysis and Security Engine (BASE) Setup Program

Step 1 of 5

Pick a Language: portuguese [?]

Path to ADODB: /etc/apache2/adodb5 [?]

Continue

Fonte: próprio autor.

Na Figura 23, podemos ver a configuração do Banco de Dados do BASE. Nesse ponto, deverá ser configurado o banco e o usuário que foram criados nos passos anteriores, referentes ao MySQL e o Barnyard2.

Nome do Banco: **db_snort**

Servidor MySQL (Host): **localhost**

Nome do Usuário: **usr_snort**

Figura 23: Configuração do Banco de Dados

Basic Analysis and Security Engine (BASE) Setup Program

Step 2 of 5	
Pick a Database type:	MySQL [?]
Database Name:	db_snort
Database Host:	localhost
Database Port: Leave blank for default!	
Database User Name:	usr_snort
Database Password:	●●●●●●
<input type="checkbox"/> Use Archive Database[?]	
Archive Database Name:	
Archive Database Host:	
Archive Database Port: Leave blank for default!	
Archive Database User Name:	
Archive Database Password:	
Continue	

Fonte: próprio autor.

Configurações de autenticação. No projeto foi utilizado o usuário **admin**.

Figura 24: Configuração de usuário para administração

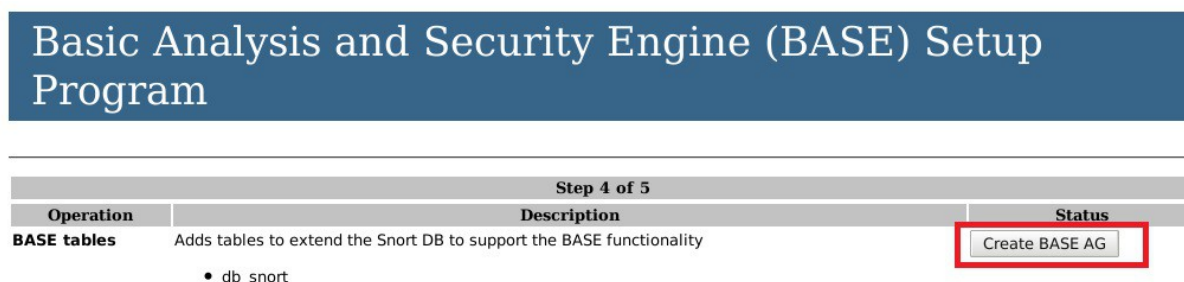
Basic Analysis and Security Engine (BASE) Setup Program

Step 3 of 5	
<input type="checkbox"/> Use Authentication System [?]	
Admin User Name:	admin
Password:	●●●●●●
Full Name:	Administrador
Continue	

Fonte: próprio autor.

Criação das tabelas no banco de dados.

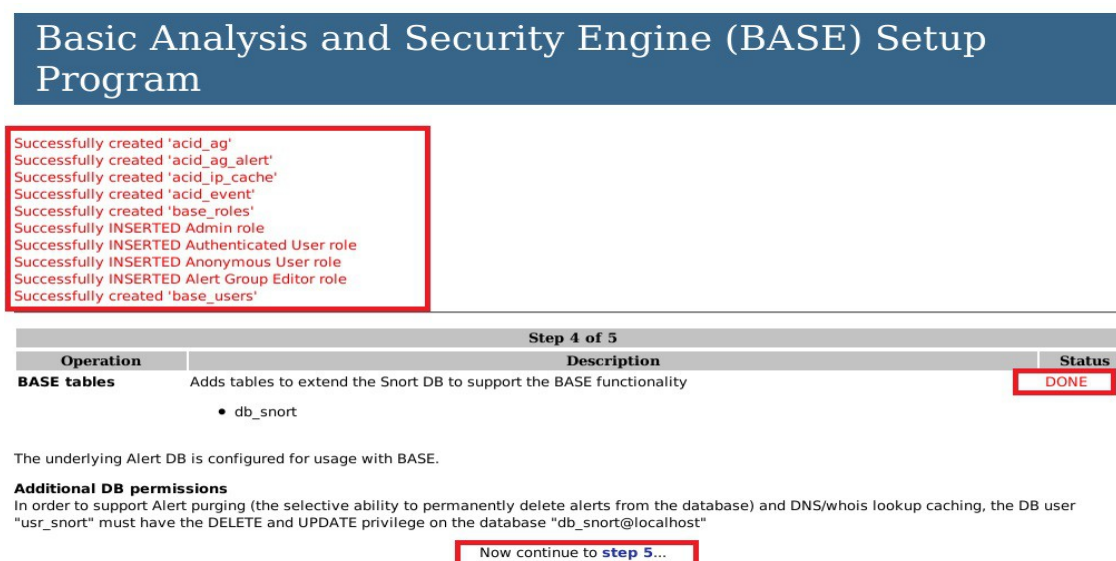
Figura 25: Criação das tabelas (BASE).



Fonte: próprio autor.

Se tudo estiver correto, a tela mostrada na Figura 26 deverá ser mostrada. Clique em “Now continue to step 5...” para continuar com a configuração.

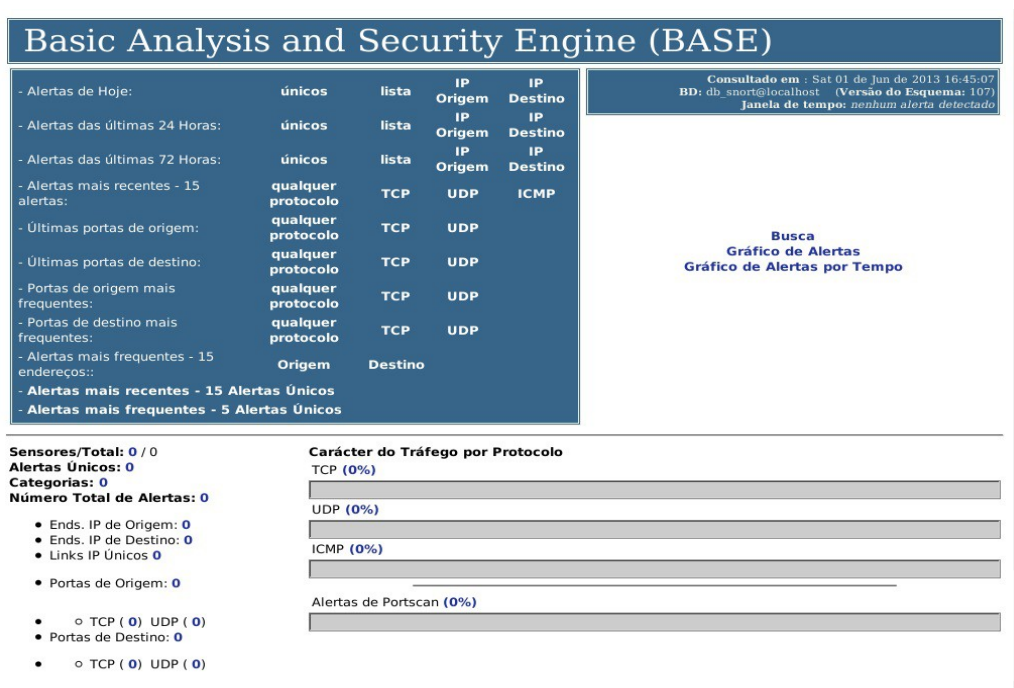
Figura 26: Processo de criação das tabelas



Fonte: próprio autor.

Tela principal do BASE, onde mostra a situação do ambiente, como a quantidade de sensores, total de alertas entre outras informações que podemos ver na imagem abaixo.

Figura 27: Tela principal do BASE - Monitoramento



Fonte: próprio autor.

Com a instalação do BASE finalizada, será preciso voltar ao *Firewall*, para que possamos alterar as permissões do diretório e deixá-las da forma correta.

Comando para alterar as permissões do diretório:

```
chmod -R 755 /var/apache2/htdocs/base/
```

```
find /var/apache2/htdocs/base/ -type f | xargs chmod 644
```

Será preciso alterar um dos arquivos de configuração do BASE, chamado **base_conf.php**. Essa alteração é importante, para que a interface web seja atualizada automaticamente em um intervalo de tempo mais curto, possibilitando que os dados referentes aos ataques, sejam visualizados instantaneamente.

Edite o arquivo:

```
/var/apache2/htdocs/base/base_conf.php
```

Localize a linha:

```
$stat_page_refresh_time = 180;
```

e modifique-a para:

```
$stat_page_refresh_time = 2;
```

A 2.5 Instalação e configuração do snort.

Antes de iniciar a instalação do SNORT, será preciso instalar suas dependências.

PCRE - Biblioteca de expressões regulares.

```
pkg_add pcre
```

LIBDNET – API genérica de rede, que possibilita o acesso a vários protocolos.

```
pkg_add libdnet
```

DAQ – API de aquisição de dados

```
pkg_add daq
```

Instalação e configuração do SNORT

```
pkg_add snort
```

Editar o arquivo de configuração do snort: /etc/snort/snort.conf

```
/etc/snort/snort.conf
```

Localizar a linha:

```
# Official Sourcefire VRT rules from http://www.snort.org/snort-rules/
```

Em seguida, deve-se comentar ou excluir todas as linhas abaixo desta, cujo conteúdo comece com:

```
include $RULE_PATH/nome-da-regra
```

E deixar apenas a regra que será utilizada.

```
include $RULE_PATH/dos.rules
```

Caso esse procedimento não seja realizado, o processo do SNORT não irá iniciar, pois não vai conseguir encontrar todas as regras que estão listadas dentro do arquivo de configuração.

OBS: *Essas regras podem ser baixadas ou criadas posteriormente, mas para o ambiente de teste elas não serão utilizadas.*

Ainda no arquivo de configuração do SNORT, modificar as linhas abaixo:

```
output unified2: filename snort.u2, limit 128
config logdir: /var/log/snort/
```

Criar regra para detectar o ataque.

Vamos precisar criar o arquivo que contém a regra de detecção do ataque. Nesse projeto, será utilizado apenas uma regra, criada especialmente para detectar o ataque que foi simulado. Provavelmente uma outra regra, escrita de forma um pouco diferente, também conseguiria detectar o ataque. A regra foi criada na base da “tentativa e erro”, após estudar outras detecções de ataques de negação de serviço (TRUSTWAVE'S SPIDERLABS) e simulando o ataque várias vezes, analisando o tráfego capturado através do TCPDUMP para conseguir entender seu modo de operação.

Criar o arquivo de regras

```
touch /etc/snort/rules/dos.rules
```

Editar o arquivo criado e inserir as linhas abaixo:

```
alert tcp any any -> any any (msg:"Possível Ataque de Negação de Serviço";
threshold: type threshold, track by_src, count 100 , seconds 5; classtype:misc-activity;
sid:2013; rev:1;)
```

A linha **alert tcp any any -> any any**, faz com que o SNORT analise o tráfego que entra ou sai de qualquer interface de rede, permitindo que o ataque seja detectado mesmo que este venha da rede interna.

A linha **threshold: type threshold, track by_src, count 100 , seconds 5**, significa 100 conexões a cada 5 segundos, ou seja, quando o SLOWLORIS começar a agir, ele vai justamente realizar várias conexões ao servidor em um intervalo de tempo bastante curto, o que conseqüentemente irá fazer com que o o SNORT gere um alerta imediatamente.

Criar diretório de logs:

```
mkdir -p /var/log/snort/
```

Para que o SNORT funcione da forma que precisamos, ou seja como um NIDS, vamos precisar alterar o script de inicialização, localizado em `/etc/rc.d/`.

Edite o arquivo de inicialização do SNORT:

```
/etc/rc.d/snort
```

Apague todo o conteúdo do arquivo e insira os dados abaixo:

```
#!/bin/sh
#
daemon="/usr/local/bin/snort -D"
daemon_flags="-dev -c /etc/snort/snort.conf -u root"

. /etc/rc.d/rc.subr

rc_cmd $1
```

Caso essa alteração não seja realizada, haverá problemas com o serviço do Barnyard2, que não vai conseguir ler os logs do SNORT, o que também irá afetar o BASE.

Com todos os serviços instalados e configurados, precisamos configurar o sistema para que estes serviços sejam carregados automaticamente durante a inicialização. No OpenBSD, o arquivo responsável pela inicialização é o **/etc/rc.conf**, porém, de acordo com a documentação do OpenBSD, não é aconselhável editar esse arquivo diretamente, pois após uma atualização esse arquivo poderá ser modificado automaticamente, causando problemas mais tarde. Para fazer a configuração de forma correta, vamos criar o arquivo **rc.conf.local**, dentro do diretório **/etc**. Esse arquivo será o responsável por iniciar os serviços que precisamos automaticamente durante a inicialização do sistema.

Criar o arquivo “rc.conf.local”:

```
touch /etc/rc.conf.local
```

Edite o arquivo criado:

```
/etc/rc.conf.local
```

Insira os dados abaixo:

```
pkg_scripts="mysqld httpd2 snort barnyard2"
```

A 3 Configuração do servidor web

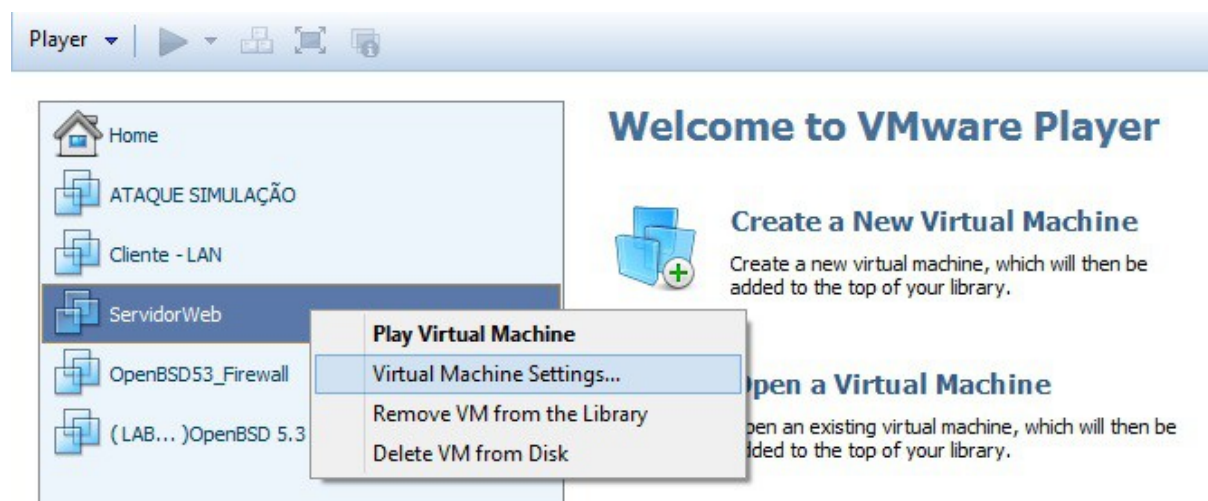
Este trabalho acadêmico não irá abordar a instalação e configuração de um servidor web, devido ao fato deste procedimento fugir do escopo do projeto.

Como explicado anteriormente, o servidor web também será uma máquina virtual. O sistema operacional instalado foi o Linux CentOS 6.4 (32bits) e nesse servidor foi instalado o Apache (versão 2.2.15), disponível nos repositórios oficiais do CentOS.

Abaixo veremos como deve ficar a configuração do VmWare Player referente ao servidor web.

Clique com o botão direito do mouse sobre a máquina virtual e selecione a opção “Virtual Machine Settings...”.

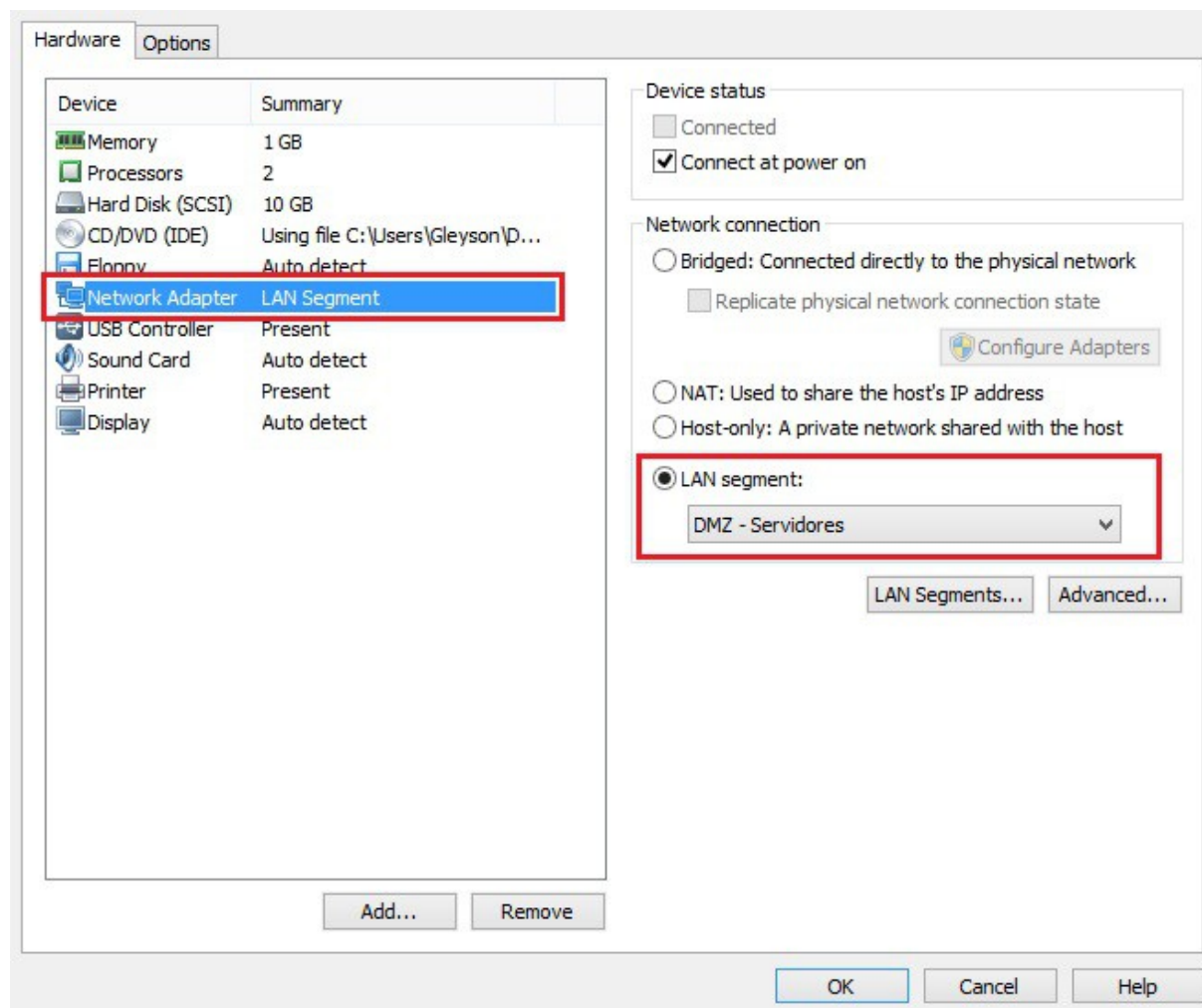
Figura 28: Propriedades da Máquina Virtual



Fonte: próprio autor.

A interface de rede do servidor web, deverá estar no segmento de rede destinado a DMZ (DMZ – Servidores). A imagem abaixo mostra como a configuração deve ser realizada.

Figura 29: Configuração da interface de rede



Fonte: próprio autor.

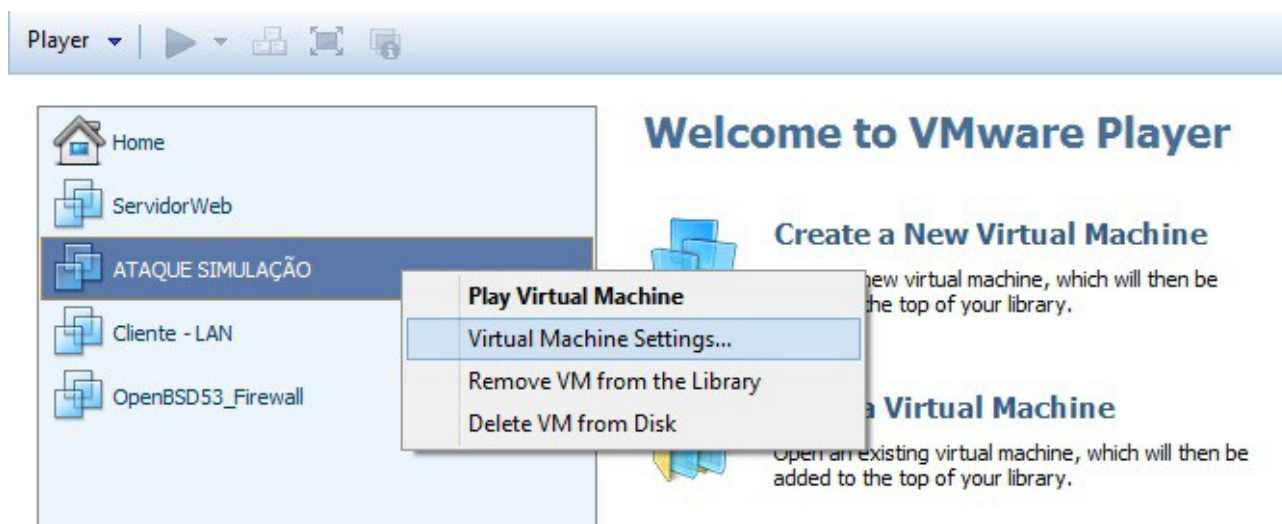
A 4 Configuração da estação do atacante

A estação do atacante poderia estar funcionando com qualquer distribuição Linux. Nesse projeto, optou-se pelo GNU/Linux Debian 6.0 (32bits). A ferramenta de ataque escolhida para simular o ataque foi o Slowloris, devido a sua simplicidade, porém grande eficácia.

Também não será abordada a instalação do sistema operacional na estação do atacante, mas a instalação e utilização da ferramenta de ataque será devidamente explicada.

Clicar com o botão direito do mouse sobre a estação do atacante. Selecionar o opção “Virtual Machine Settings...”.

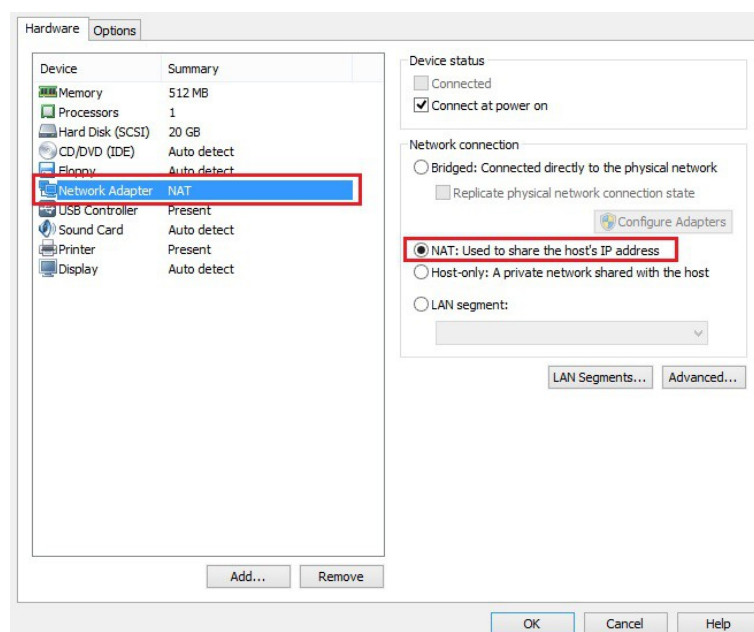
Figura 30: Configuração Estação do Atacante



Fonte: próprio autor.

A interface de rede deverá estar configurada para funcionar em modo “NAT”. A Figura 31 mostra como deve ser realizada a configuração da interface de rede.

Figura 31: Configuração da interface de rede



Fonte: próprio autor.

A 4.1 Instalação do SLOWLORIS HTTP DOS

Antes de executarmos o Slowloris, será preciso instalar algumas dependências. O Slowloris é apenas um pequeno *script*, escrito em linguagem Perl, então é necessário que haja suporte para esta linguagem. Nesse projeto, como foi dito anteriormente, será utilizado a distribuição GNU/Linux Debian. Em uma instalação padrão, o suporte a linguagem Perl já está incluída, restando apenas a instalação de mais algumas dependências.

Instalação de dependências

```
apt-get install libio-socket-ssl-perl
```

O Site do desenvolvedor, recomenda os passos abaixo, apesar do comando anterior resolver o problema das dependência.

```
perl -MCPAN -e 'install IO::Socket::INET'
```

```
perl -MCPAN -e 'install IO::Socket::SSL'
```